

User Group Analytics Survey and Research Opportunities

Behrooz Omidvar-Tehrani, Sihem Amer-Yahia

▶ **To cite this version:**

Behrooz Omidvar-Tehrani, Sihem Amer-Yahia. User Group Analytics Survey and Research Opportunities. IEEE Transactions on Knowledge and Data Engineering, 2020, 10.1109/TKDE.2019.2913651 . hal-02972499

HAL Id: hal-02972499
https://hal.science/hal-02972499

Submitted on 29 Nov 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

User Group Analytics Survey and Research Opportunities

Behrooz Omidvar-Tehrani, *Member, IEEE,* and Sihem Amer-Yahia, *Member, IEEE*

Abstract—User data can be acquired from various domains and is characterized by a combination of demographics such as age and occupation, and user actions such as rating a movie or recording one's blood pressure. User data is appealing to analysts in their role as data scientists who seek to conduct large-scale population studies, and gain insights on various population segments. It is also appealing to users in their role as information consumers who use the social Web for routine tasks such as finding a book club or choosing a physical activity. User data analytics usually relies on identifying group-level behaviors such as "Asian women who publish regularly in databases". Group analytics addresses peculiarities of user data such as noise and sparsity to enable insights. In this survey, we discuss different approaches for each component of user group analytics, i.e., discovery, exploration, and visualization. We focus on related work which arises from combining those components. We also discuss challenges and future directions of having an all-in-one system, where all those components are combined. This survey has been presented in the form of two tutorials [1], [2].

Index Terms—User data, user group, user group analytics.

1 Introduction

User data is of interest to analysts in their role as data scientists who seek to conduct large-scale population studies, and gain insights on various population segments. It is appealing to users in their role as information consumers who use the social Web for routine tasks such as finding a book club or choosing a restaurant. It is also useful to domain experts who seek to understand their users.

We define user data analytics as a collection of methods and tools to extract value from user data. It relates to a special field of business analytics, referred to as behavioral analytics [3]. The goal of behavioral analytics is to unveil insights into the behavior of consumers on eCommerce platforms, online games, IoT and web and mobile applications. For example, in e-commerce and retail, behavioral analytics serves product recommendations and predicting future sale trends. In online gaming, predicting usage trends shapes future releases. Similarly, determining how users use an application helps predict future usage and preferences in application development.

A common way of analyzing user data is *user group analysis* whose purpose is to breakdown users into groups to gain a more focused understanding of their behavior. We refer to that as User Group Analytics (UGA). UGA helps analysts make better and faster decisions [4] with higher certainty [5]. UGA also addresses peculiarities of user data such as noise and sparsity. UGA is useful to social scientists who look to conduct large-scale population studies. UGA is also useful in forming suggestions of alike people when looking for a restaurant or a fan club [6], [7].

For the purpose of this survey, we propose to organize the different components that form UGA into a single architecture depicted in Fig. 1. UGA starts from raw user data and **discovers groups** that reflect the behavior of a set of users, e.g., "Asian women who publish regularly in databases". User data must first be cleansed and enriched [8] before group discovery (i.e., Preparation component in Fig. 1.) Once groups are discovered, there are still two challenges before turning results into insights.

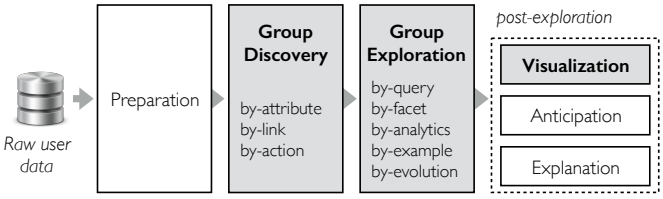


Fig. 1. User group analytics framework.

- First, there may be millions of groups. An exhaustive scan through all groups is not possible for analysts [9]. The **group exploration** component is designed to address this challenge. It operates on the results of group discovery in an iterative manner. It provides means to analysts to navigate in the space of resulting groups.
- Second, groups need to be rendered in a human-understandable form. To tackle this challenge, a post-exploration layer is designed to include auxiliary modules for rendering exploration results, including mapping them to visual variables [10], [11], [12], anticipating content and action recommendations [13], and explaining groups [14]. Explanations are provided for anomalies, outliers and interventions [15], [16], [17]. Although some anticipation will be discussed in this survey, our focus will be on **group visualization**. When analyzing aggregated data (such as user groups), human brains perform better on visual elements than textual information [4].

In this survey, we review the related work on discovery,

- ***Affiliation.*** *Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG, F-38000 Grenoble France / Institute of Engineering*
- ***E-mail.*** *firstname.lastname@univ-grenoble-alpes.fr*

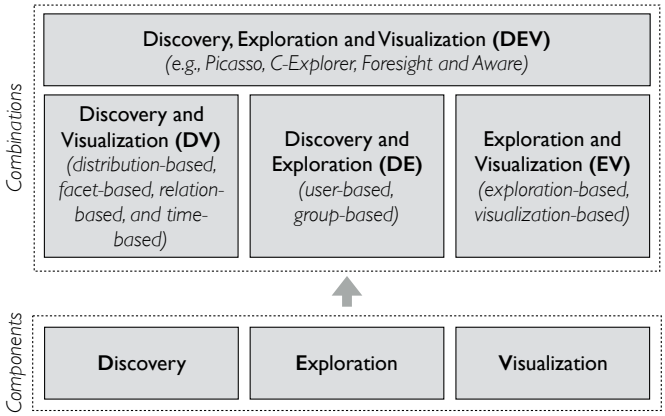


Fig. 2. UGA components and their combinations.

exploration and visualization of user groups. These components are highlighted in Fig. 1.

As analysts are empowered with expressive tools, their user data analysis needs grow. As a result, an ideal UGA system should collectively benefit from the functionalities of all its components. While a UGA task can be formulated only in terms of a single component, recent applications often require to combine some of those components. In this survey, we first review related work in UGA for each component separately and then focus on their combinations. We then discuss recent work which brings all the pieces together, and the challenges behind building an all-in-one UGA. Fig. 2 illustrates different components of UGA and their combinations.

Various evaluation protocols are proposed separately for each UGA component. Group discovery is often evaluated using efficiency measures such as response time and memory needs. Visualization is usually measured qualitatively by a user study. Exploration evaluation is in-between and often employs both qualitative and quantitative measures. In Section 5.4, we review different evaluation strategies for each UGA component and their combinations. We also discuss future directions of designing an evaluation approach for an all-in-one UGA.

The survey is organized as follows. In Section 2, we provide a generic model for user data and user groups. Then we review *past* (single components of UGA), *current* (pairwise combinations of UGA components), and *future* trends (all-in-one UGA) in user group analytics. Section 3 describes the past by covering related work for each individual UGA component. Section 4 discusses the current status of UGA frameworks and focuses on combining UGA components. Then in Section 5, we discuss the future of UGA and present challenges and opportunities of building an all-in-one UGA system. We conclude in Section 6. We note that this survey has been presented in the form of two tutorials [1], [2].

2 User Group Model and Use Cases

Given a set of users $\mathcal{U}$ and a set of items $\mathcal{I}$, we define *user data* as a database $\mathcal{D}$ of tuples $\langle u, i, v \rangle$ which represent a value $v \in \mathbb{R}$ induced by an action such as browsing, tagging, or rating, of user $u \in \mathcal{U}$, on item $i \in \mathcal{I}$. The notation used in this paper is summarized in Table 1.

TABLE 1
Notation reference

$\mathcal{D}$	User dataset	A	Attributes
$\mathcal{U}$	Users	ρ	Discovery objective
$\mathcal{I}$	Items	θ	Exploration type
$\mathcal{G}$	User groups	$\mathcal{V}$	Visual variables

This generic data model describes multiple user datasets in the literature, such as $1.7M$ research publishing actions of database researchers[1] [18], $5B$ tweets [19], $300M$ customer receipts from a retail chain of 1,800 stores [20], $10M$ rating records from MovieLens[2] [21], $50M$ artist ratings from LastFM[3] [22], $1M$ electronic health records (EHR) [23] and about $200K$ book ratings from BookCrossing[4] [24]. For instance in MovieLens, the tuple $\langle John, Titanic, 4 \rangle$ describes that the user John rated the movie Titanic with a score of 4. Also for the tweets dataset, the tuple $\langle Tiffany, tweet_id, Hemophilia \rangle$ describes that Tiffany tweets about Hemophilia.

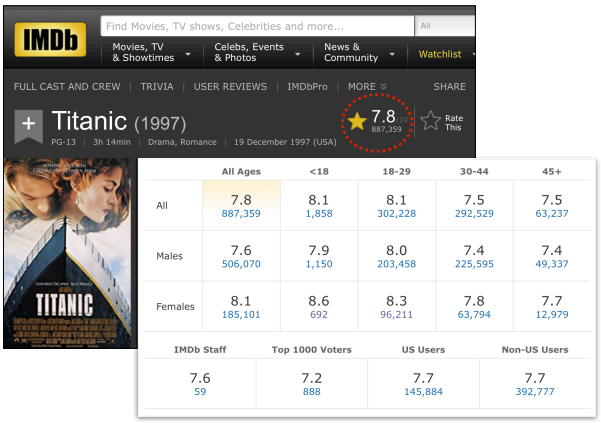


Fig. 3. Pre-defined user groups in IMDb.

Attributes. Users and items have attributes drawn from a set A. Instances of user attributes are age, gender, diet, political orientation, and occupation. Instances of item attributes are book author, movie director, tweet language, and treatment duration. The choice of what constitutes an attribute is application-dependent. For instance, a user on a collaborative rating site may have an "age" and "gender", and a book has an "author" and "publisher".

Social links. A social link $\langle u, u', link_type \rangle$ depicts the potential bond between a pair of users u and u' where $link_type$ may represent co-authorship [25], affinity [26] or friendship [27].

2.1 User Groups

User data analytics is often conducted via *user groups*. Data scientists conduct large-scale population studies and gain

1. *DM-Authors dataset: http://dx.doi.org/10.18709/PERSCIDO.2016.10.DS32.*
2. *MovieLens dataset: https://grouplens.org/datasets/movielens/.*
3. *LastFM dataset: https://labrosa.ee.columbia.edu/millionsong/lastfm.*
4. *BookCrossing dataset: https://grouplens.org/datasets/book-crossing/.*

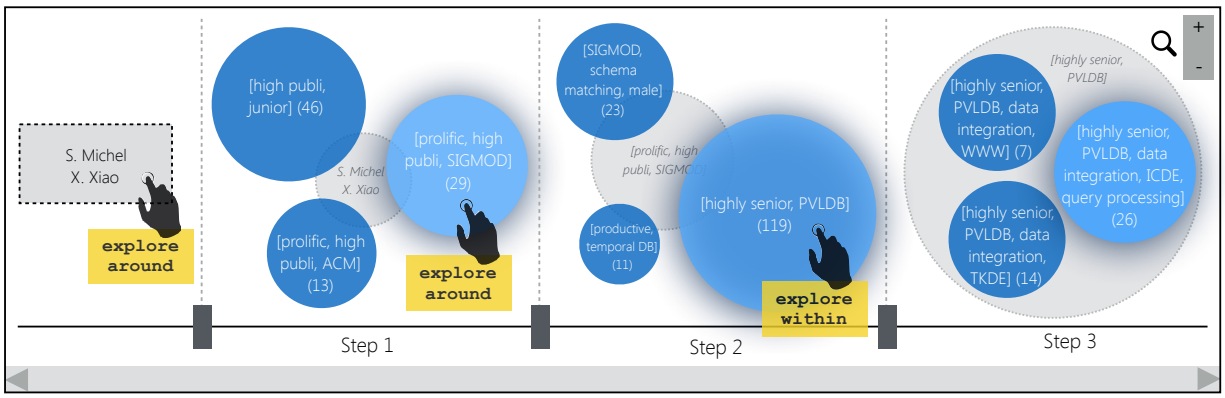


Fig. 4. UGA use case for domain experts: WebDB 2014 program committee formation.

insights on the preferences of different population segments. Information consumers explore alike user groups to be inspired for routine tasks such as choosing a restaurant or picking a TV subscription. Domain experts need to understand their users and their actions to be able to predict future actions and detect abnormal behavior. Fig. 3 illustrates precomputed user groups proposed in IMDb[5] for the movie Titanic. This breakdown shows groups like *female teenagers* and *young female reviewers* whose rating average differs from the overall average (8.6 and 8.3 respectively, versus the overall average of 7.8).

A user group g is a subset of $\mathcal{U}$ whose members have *common attributes* (e.g., having the same gender and occupation) and *common actions* (e.g., watching the same set of movies). For instance a user group $g = [\langle$gender, female$\rangle$, $\langle$age, young$\rangle$, Titanic] describes young female users who rated the movie Titanic. Throughout this survey, whenever it is clear from the context, we use the word *group* instead of user group. Also, when attributes are clear from the context, we will show group labels in a more concise form, e.g., $g = [$female, young, Titanic$]$.

We use $\mathcal{G}$ to denote the set of all user groups. $|\mathcal{G}|$ is equal to the size of the powerset of attribute values and items. For instance, with $|\mathcal{I}| = 5$, $|A| = 4$ and 3 values per attribute, $|\mathcal{G}|$ will be $2^{(5+4\times3)}$ which is in the order of 10^5.

In the literature, user groups have been referred to with different terms, such as *communities* [28], [29], *tribes* [30], *cliques* [31], *cohorts* [32], *teams* [33], *segments* [34], *patterns* [35], [36], *cubes* [13], *clusters* [37], [38] and *partitions* [39]. For instance, clusters and partitions are assumed to represent non-overlapping user groups. Teams are sometimes used to refer to groups with one member as a leader. Communities are considered as groups whose social links are more important than their attributes.

2.2 Use Cases

We present few use cases which illustrate different applications of UGA and its components in real world. Each use case captures a specific kind of users, i.e., data scientists, domain experts, and information consumers.

5. *The Internet Movie Database (IMDb): http://www.imdb.com*

Use Case 1 (Telecom Customers). Suzanne is a data scientist in a telecom company. She wants to **discover** groups of customers with common stories as ad targets. To find interesting groups, she **explores** similar ones. Then, she **visualizes** a subset of interest for visual inspection. She finds two promising groups, i.e., *all-day customers requiring 2 SIM cards for the same smartphone* and *customers who mainly use their phone during breaks*. Those will constitute her target for ads.

Use Case 2 (Expert-Set Formation). Martin is WebDB 2014 program chair[6] (i.e., a domain expert). He wants to build a program committee formed by geographically distributed male and female researchers with different seniority and expertise levels. He **explores** various groups of researchers **visually** (Fig. 4). Starting from a seed set, the system **discovers** few groups out of which the group described as [prolific, high publications and publishing in SIGMOD], has 29 geographically-distributed and gender-distributed researchers. Martin chooses few candidates for the committee and proceeds to the next iteration to find other members.

Use Case 3 (Quantified-Self). Mary, an information consumer, has over 100 ratings on BookCrossing. She is looking to join an online book club where she can engage in stimulating debates. Mary needs to **discover** and **explore** groups with whom she highly agrees or disagrees, i.e., those whose rating distribution is very close or very far to hers. As there exists one thousand groups within her criteria, she **visualizes** those groups to achieve a big picture for a better comparison.

3 UGA Components

UGA can be performed along three different components: *discovery*, *exploration*, and *visualization* [40]. A UGA task may be defined on each component individually or on a combination of components. An analyst may opt to focus on groups mined from raw user data (Section 3.1). Another may want to explore those groups further (Section 3.2). Another one may decide to visually analyze groups (Section 3.3). In this section, we review those components.

6. *http://webdb2014.eecs.umich.edu*

3.1 User Group Discovery

User group discovery refers to a set of approaches which derive value from user data in the form of user groups. For instance, discovery helps Suzanne, in Use Case 1, find groups as ad targets. The challenge in group discovery is the ever-increasing complexity of user data: it is not clear apriori which entities in user data (attributes, links, actions) should be leveraged to form groups.

3.1.1 Discovery process

Discovery is a function $discover(\mathcal{D}, \rho) \rightarrow \mathcal{G}$ which admits as input the user data $\mathcal{D}$ and an objective ρ, and returns the set of groups $\mathcal{G}$ where $\rho(\mathcal{G}, \mathcal{D})$ is optimized. A discovery approach can be as straightforward as a SQL query (i.e., a group-by statement), or as sophisticated as tracking evolutions in temporal user groups. In [41], [42], several discovery objectives are listed.

3.1.2 Types of group discovery

User groups are discovered using common attributes, links, and actions of users.

Attribute-based discovery. Such discovery methods consider users as individual entities and leverage their common attributes to form groups. Representative discovery aims to mine groups that best represent a subset of users [43]. In [44], [45], a multi-objective group discovery method is proposed to optimize several objectives simultaneously, such as coverage of users, homogeneity of actions, and diversity of groups. In [46], an LSTM-based discovery method is proposed to discover patients' sub-cohorts and facilitate therapeutic intervention.

Link-based discovery. Such discovery methods leverage explicit connections between pairs of users, such as affinity, following, and friendship. Social discovery is a body of work which leverages social links to discover user groups in form of communities [47], [48], [49], [50], [51], [52]. This means that the user data is divided into communities, such that users within the same community tend to be connected by links, while those within different communities tend not to be connected. One common objective is Newman's *modularity* [28] with the intuition of forming communities with stronger internal connectivity than external connectivity. Another common objective is the *density* of social links in communities. In [53], a hierarchy of dense user communities is mined. Inversely, tenuous communities are mined in [54].

Various application-dependent objectives are also proposed in the literature which exploit social links. CRAQ leverages social links as *collaboration probabilities* and maximizes the collective informative power of Twitter user groups for answering a question on the platform [55]. ENGTFP (Engaging Team Formation Problem) leverages social links as *engagement capacity values* and maximizes this capacity in an online social network [33]. LDMS leverages social links as *affinities* and maximizes a score (LinkedIn Decision Maker Score) to identify groups of "sales decision makers" in LinkedIn[7] [56]. Also in [57], social links are considered as *social closeness* and the total social distance from a group initiator (leader) is minimized.

7. *http://linkedin.com*

Action-based discovery. Action-based discovery mines groups based on their common actions [37]. The most common objective in this category is *support*, which counts the number of actions that constitute a group [38]. In case user actions are associated to *time* and *location*, time-based and location-based discovery are proposed, respectively.

Time-based discovery. In [58], [59], [60], [61], the evolution of group actions (e.g., transition of trending topics, patterns of group appearance and disappearance) is mined. Also in [62], a group with an exceptional transition behavior is discovered (e.g., US users are more likely to listen to Reggae after listening to World music). The common objective in time-based discovery is *drift*, i.e., the sudden change in user actions.

Location-based discovery. Urbanity [63] models the dynamics of user's actions in urban environments and discovers spatiotemporal hotspots of where group actions concentrate using a *spatial density* objective.

3.1.3 Discovery alone is not enough for UGA

Discovery alone is the method of choice for UGA when there is a clear question in the mind of the analyst to be formulated as an optimization problem. However, there are cases where group discovery alone is not enough. For instance, it is nearly infeasible for Suzanne in Use Case 1 to inspect all discovered groups of customers one by one to find groups of interest for advertisement. Since discovery is a one-shot process, it is incapable of handling exploratory scenarios where the analyst's question evolves and improves in time. In such cases, the analyst needs to interact with found groups to validate different hypotheses. In other words, there is a need for the analyst to be inside the analytics loop.

3.2 User Group Exploration

Exploration refers to a set of approaches which enable interaction with user groups. It is part of a trending research direction called Human-in-the-Loop Data Analytics (HILDA) where people are involved in the data analytics process. Exploration helps analysts navigate the space of user groups (i.e., $\mathcal{G}$) to obtain insights and validate their hypotheses [64]. For instance, exploring groups in a social network helps discover influential tags [65]. The challenge in group exploration is information overload, i.e., the space of all possible groups is huge, and the human cognitive perception capacity is limited [9].

In many exploration scenarios, the analyst only has a *partial understanding of her needs* and seeks to refine them as she extracts more insights from the data. For instance, Martin in Use Case 2 has a vague goal in mind (i.e., constructing a PC) and exploration helps him follow his directions of interest on researcher groups to progressively build the committee. Hence it is crucial to exploit the analyst's feedback to improve exploration.

3.2.1 Exploration process

Each analysis session may contain a sequence of iterations whose transition is formed by exploration. Exploration is a function $explore(g, \theta) \rightarrow \mathbb{P}(\mathcal{G})$ which admits as input a group $g \in \mathcal{G}$ and an *exploration type* θ and returns a set of

groups ($\mathbb{P}(\mathcal{G})$ is the power-set of $\mathcal{G}$). The exploration type specifies the analyst's needs that the exploration should be based upon.

3.2.2 Types of group exploration

Exploration is a navigation in the space of user groups. At each iteration, the analyst increases her partial understanding of the analysis task. This awareness can be captured in different forms, such as queries, facets, distributions, examples, and evolutions. According to different ways of capturing the analyst's needs, we recognize the five following exploration types: by-query, by-facet, by-analytics, by-example, and by-evolution.

By-Query. The parameter θ consists of predicates on attributes and items which form a query. In each iteration of the analysis session, the analyst formulates a query and the exploration system returns groups which satisfy the query predicates. The analyst will then update her query iteratively with the new acquired knowledge in previous iterations and the system provides other groups in line with the analyst's need [66], [67]. Note that query formulation requires a knowledge of the dataset and the query language, which is not always the case. Implementations of by-query exploration are as follows.

Explore next returns next possible groups for a given query. In [66], [68], a query is formulated in terms of constraints, and the next available group satisfying the constraints will be returned. In [69], the next random group is returned which is in line with feedback constraints of the analyst.

Explore best region forms a query based on location attributes. In [65], [70], a spatial rectangle is given as query constraints and all groups within that rectangle are returned. There exist many applications for the *explore best region* implementation, including location-based advertisement targeting [63] and signage localization for marketing a new product [70].

Online community search. In [71], [72], [73], the problem of online community search is addressed, where the query is a set of users, and the task is to find a densely connected subset of $\mathcal{U}$ that contains those users.

By-Facet. The parameter θ consists of attribute-value pairs (i.e., facets) which appeal interesting to the analyst. The exploration system returns groups whose members satisfy requested facets. For instance, if $\langle$gender, female$\rangle$ is given as an attribute-value pair in θ, a group of female students would be a good candidate for exploration. In the next iterations, the analyst adds or removes facets to receive new groups. Comparing to by-query exploration, by-facet exploration reduces the burden on the analyst specifically on datasets with many attributes [74], [75]. One implementation of by-facet exploration is *Split* which admits as input an attribute $a \in A$ and returns one group per a's value [13], [76]. The union of returned groups cover all users and the pairwise intersection of those groups yields an empty set. This by-facet implementation is equivalent to *drill-down* function in OLAP [13].

By-Analytics. The parameter θ consists of a desired distribution of user actions. The exploration system returns groups whose distribution is similar to the input. For instance in [77], a desired histogram of rating scores is given and k groups with similar rating score distributions are returned. By-analytics exploration requires a knowledge of the user data and its underlying distributions. Implementations of by-analytics exploration are as follows.

Summarize returns groups with summarized input distributions. In [78], two groups are given as input, and a group with common distribution characteristics of input groups is returned. For instance, if the members of both input groups tend to rate with low scores, a group with an average score of 2.0 (out of 10) would be returned as their summary.

Redescribe returns a group with an identical distribution with the input group but different description [79]. For instance, if an increasing distribution (i.e., less ratings for the lower scores and more ratings for the higher scores) of the movie Titanic is described with the group [female, young], it can be redescribed by the group [female, teenager], as both groups rate Titanic in the same way (see Fig. 3).

Contrast returns a group whose distribution is totally different from the given group [80]. For instance, Martin in Use Case 2 may be interested to bring diversity in PC selection by exploring researcher groups who are entirely different from the ones he is currently investigating on.

By-Example. The parameter θ consists of group examples. The analyst provides examples of what she needs to get and the system explores other groups similar to those provided examples. Example-based exploration is beneficial where the analyst is not able to express her needs otherwise [40], [81]. Implementations of by-example exploration are as follows.

Remove prunes $\mathcal{G}$ by removing all groups similar to the given group [36], [82].

Neighbors returns k nearest neighbors of the given group [29], [83].

Explore around returns k similar groups to the input group with maximal diversity (i.e., they are as distinct as possible from each other). Also *Explore within* returns k sub-groups of the input group with maximal coverage [18] (see Fig. 4).

By-Evolution. The parameter θ consists of a group and a desired point in time (timestamps, period, season, semester) and the system returns the evolved status of the group of interest at that time. An implementation of by-evolution is [58] where analysts can observe the evolution of communities over time.

Which exploration type to choose? Data scientists often choose by-query exploration to quickly discover subsets of interest. However, it is only possible in case of a full knowledge of the users and the analysis task. Domain experts have often a good understanding of their data but they do not know how to form queries. Their method of choice is then by-facet, by-evolution, or by-analytics exploration. Information consumers prefer to use by-example exploration. If they know domain attributes (e.g., in case of Amazon or eBay electronic commerce sites), then by-facet exploration can also be considered. However, with new evolving needs for UGA, the aforementioned categorization

TABLE 2
Mapping visual variables to group characteristics

Visual variables $\mathcal{V}$	**Mappings $\mathcal{V}(G)$**
Circle (Shape)	Group
Size	Number of members
Line	Between two groups in case they are related
Line width	Number of users in common between a pair of groups
Color intensity	Value of an ordinal attribute (e.g., age, education)
Distinct colors	Value of a categorical attribute (e.g., gender, occupation)

between different roles is fading away and different roles may leverage different exploration strategies.

3.2.3 Exploration alone is not enough for UGA

Exploration enables a natural dialog between analysts and user groups. Enabling interaction provides customization (i.e., better understanding of the analyst's partial needs) and a better handling of heterogeneity [84]. When the question is not clear in the mind of the analyst, exploration is the method of choice for UGA as it enables iterative exploratory scenarios. However, there are cases where exploration alone is not enough. First, the result of exploration is not necessarily designed to be fully comprehensible and readable by analysts. Second, since user data is voluminous, noisy and sparse, direct interactions with it may not easily and quickly lead to insights. For instance, Mary in Use Case 3 cannot make sense of similar and dissimilar groups without visualizing them, because she cannot easily compare groups with each other.

3.3 User Group Visualization

Visualization refers to a set of approaches which enable sensemaking of user groups using visual variables [85]. It adds value to insights with the use of visual views rather than textual or tabular content [86]. The challenge in group visualization is clutteredness, i.e., numerous overlapping groups make it almost infeasible to visually make sense of groups. A visualization component consists of the following building blocks: views, visual variables, and visual elements.

- *View.* A view contains one possible visual representation of user groups. The representation may be in the from of a histogram, a pie-chart or separated clusters. In many scenarios, analysts may be provided more than one view for a set of user groups to inspect various aspects of their analysis goal.
- *Visual variable.* A representation is formed by one or several visual variables (encodings), such as lines, ticks, colors and shapes. Obviously, visual variables alone do not carry any semantics. We denote visual variables as $\mathcal{V}$.
- *Visual element.* An instantiation of a visual variable is called a visual element. Each visual element carries a semantics. For instance, an element leverages *size* (i.e., a visual variable) to illustrate the number of group members. For instance in Fig. 4, user groups are illustrated using overlapping circles.

3.3.1 Visualization process

At the core of visualizing user groups sits a mapping function $visualize(G \subseteq \mathcal{G}) \rightarrow \mathcal{V}(G)$ which associates G's characteristics (size, members in common, description, etc.) to visual variables, i.e., $\mathcal{V}(G)$ in each view. Table 2 contains common visual mappings for user groups. The $visualize()$ function is responsible to set scales, projections, axes, legends and marks [87]. A visualization approach may opt for visualizing groups with their members or as atomic concepts.

3.3.2 Types of group visualization

The literature contains very few approaches for visualizing user groups. Traditionally, this is performed directly on raw user data using off-the-shelf visualization products and libraries such as Tableau[8], Spotfire[9], QlikView[10], Gephi[11], D3[12], OpenGL[13] and HTML Canvas (SVG). Applied directly on raw data, these solutions are mostly static and do not fully support sophisticated views on user groups. In the following, we review few classes of group visualization approaches as improvements to the status quo.

Graph visualization. User group visualization inherits a long history of graph visualization where nodes are either users or groups, and edges are either social links or relation between groups (e.g., common users and common actions), respectively [88], [89], [90]. Note that in case social links are not available, graph visualization is not functional.

Time-based visualization. In case user actions are time-stamped, evolution of actions and groups can be visualized. Event analytics tools such as [23], [82], [91] visualize temporal actions of group members as visual trends. While such methods perform well to convey a big picture on trends, often having many heterogeneous actions with irregular times results in a cluttered and unusable view.

Location-based visualization. In case user actions are localized, spatial visualization of latitudes and longitudes on geographical maps can be leveraged. In [92], bike sharing activities (i.e., user actions) are grouped and visualized to understand the dynamics of station states and trip circulation patterns. In map visualization, groups are often mapped to circles and distinct colors are leveraged to show different attributes and actions.

Application-dependent visualization. Tailored visualization approaches are proposed for specific applications. VisOHC [12] visualizes user groups and their actions in Online Health Communities (OHC) to better administer the community and increase the engagement of its members. Visualization is also heavily used in sports analytics where players are users and their actions are visualized to obtain insights [93], [94]. Colors and color intensities are often used to show different attributes and actions of users.

8. *Tableau software suite: https://www.tableau.com/*
9. *Tibco Spotfire: https://spotfire.tibco.com*
10. *QlikView: https://www.qlik.com/us/products/qlikview*
11. *Gephi: The Open Graph Viz Platform: https://gephi.org*
12. *D3 JavaScript library: https://d3js.org*
13. *https://www.opengl.org*

TABLE 3
Summary of **DV** approaches

Approach	Key advantage	Group viz.
Distribution-based [95], [96]	General inspection of independent groups	Only groups
Facet-based [97]	Focused inspection of groups	
Relation-based [98], [99], [100], [101]	General inspection of group correlations	Groups with members
Time-based [102], [103]	Focused inspection with temporal facets	

3.3.3 Visualization alone is not enough for UGA

Most visualizations are designed as a one-shot UGA. This impedes investigating exploratory scenarios. Even in a single view, high overlap between user groups may cause a confetti effect (i.e., overlay of visual elements representing groups). Also most visualization approaches are not integrated with a data processing backbone as the infrastructure, hence they do not scale to real-world user datasets.

4 Combining UGA Components

In this section, we review existing work where a pair of UGA components are combined to benefit from their both collective advantages. It is important to note that we do not discuss only related work specifically curated for user groups, but we also discuss work that can be transposed to manipulating user groups. The common point between all those works is in their functionality on aggregations of data (including user groups) as a first-class citizen.

4.1 Combining Discovery and Visualization

A combination of group discovery and group visualization (denoted as **DV**) enables *visual inspection of discovered user groups*. For instance, Mary in Use Case 3 visualizes various possibilities of discovered groups for a better comparison and fruitful decision making. **DV** is employed to increase the *readability* and *verifiability* of discovered groups. Analysts inspect discoveries in visual form, and if they are not satisfied enough for their task, they *rediscover* by changing parameters of the optimization objective ρ. In this section, we first introduce different types of **DV** approaches and then review related work on readability and verifiability of discoveries.

4.1.1 Types of *DV* approaches

Given a set of discovered groups $\mathcal{G}$, a **DV** approach employs different visual elements $\mathcal{V}$ to highlight different characteristics of groups in $\mathcal{G}$. Table 3 summarizes the **DV** approaches and we describe them as follows.

Distribution-based visualization. Self-Organizing Maps are employed in [95] to visualize the overall distribution of actions in discovered user groups, where the whole visual view represents one single group of interest, and actions of group members are visualized as squares. Belt charts (or Sunbursts) are used in [96] to provide a more focused view on biases in distributions (dominating attribute-values) of a single group (e.g., presence of more females in groups than males.) In case biases are not in line with the analyst's goals, she requests to rediscover.

Facet-based visualization. In case the analyst wants to focus on one specific facet of discovered groups, a 3D regression heatmap can be adapted to user groups to organize all groups in a 3D grid reflecting the extent of correlation between groups and the given facet [97]. Having a specific facet in mind, the method verifies the quality of regressions and ask for a potential rediscovery. However, in case a big picture needs to be visualized, this method is not functional.

Relation-based visualization. In [98], [99], hierarchical relations between discovered groups are visualized to provide a big picture of the group space. PIVOTSLICE focuses on one single group and visualizes relations between users in that group [100]. These relations can be explicit (i.e., social links) or implicit (common actions performed by users). It also enables few manipulation operations to visually browse different relations (e.g., grouping, aligning, sorting and filtering). PACKEDCIRCLES combines the two other techniques and enables a hierarchical visualization of groups and their members [101]. In case relations do not capture the analyst's intuitions, rediscovery will be requested.

Time-based visualization. In case user actions are timestamped, WEBCANVAS can be adapted to user groups to visualize the sequence of actions in each discovered group and facilitate verifying *what happens next* in groups [102]. Groups are considered as clusters and several parallel visual views illustrate clusters side-by-side. However, these parallel views may suffer from a confetti effect. To tackle this, a visualization approach is proposed to summarize sequences and detail them on-demand [103]. In case sequences do not reflect the analyst's goals, a rediscovery may be requested.

4.1.2 Which DV approach to choose?

For a general inspection of discovered groups, analysts employ distribution-based and relation-based visualization methods to receive a big picture of the discovery. While the former focuses on groups as independent entities, the latter depicts their correlations as well. For a focused inspection of specific attributes, facet-based visualization is employed. In case of temporal facets, time-based visualization is the choice. However, in case the visual inspection is in the particular context of readability and verifiability, methods discussed in Sections 4.1.3 and 4.1.4 should be employed, respectively. Another dimension is the granularity of visualization. In case analysts want to visualize *discovered groups alongside their members*, relation-based and time-based visualizations are proposed. In case analysts want to visualize *groups as an atomic concept*, distribution-based and facet-based visualization should be picked.

4.1.3 Using *DV* for readability

Groups are often hard to read due to the presence of many heterogeneous attributes. Traditionally, readability is addressed using scatter plots and parallel coordinates to provide a clear representation along different attributes [104], [105]. Recently, dimensionality reduction has become the method of choice to represent a clear 2D visualization of groups. The *proximity* in the 2D view reveals the similarity

between groups and their members. Popular dimensionality reduction methods are Principle Component Analysis, Multidimensional Scaling and t-distributed Stochastic Neighbor Embedding. Most dimensionality reduction methods are distribution-based.

Principle Component Analysis (PCA). The focus of PCA is to capture variance in groups [106]. Given a group, PCA uses its covariance matrix to perform a linear transform from its attributes to two new orthogonal dimensions with the largest possible variance (aka, the Rayleigh quotient). However, the linearity of PCA dismisses the similarities between group members.

Multidimensional Scaling (MS). MS focuses on finding a matching from the $|A|$-dimensional space to a 2 dimensional space which preserves similarities between group members [107]. The advantage of MS over PCA is in its extended functionality to non-linear mappings. MS minimizes a *stress* function which captures the difference of user similarities between the original view and the 2D view.

t-distributed Stochastic Neighbor Embedding (t-SNE). While t-SNE has the same manifold nature as MS, it focuses on local structures of group members to obtain a clearer view [108]. In lieu of the stress function, t-SNE minimizes the KL-divergence between the distribution of user similarities in the original view and the 2D view to separate dissimilar members even more.

Most dimensionality reduction methods suffer from two drawbacks. First, they do not scale to real-size user data. Second, they do not incorporate time of user actions. We review dimensionality reduction improvements as follows.

Efficiency in dimensionality reduction. Visual results should be rendered fast so that the analyst can quickly verify discovered groups in a readable format and request a rediscovery if needed. Most dimensionality reduction methods are designed for small datasets and hence are slow on real-size user datasets [109]. CLOUDVISTA [110] is an efficient distribution-based approach for visualizing groups as multidimensional clusters. It performs dimensionality reduction on *visual frames*, i.e., a subset of user data bounded to the resolution of the visualization. The size of visual frames is much smaller than the original data, hence faster to compute. Visual frames are also parallel in nature and can be easily distributed among machines following typical MapReduce approaches.

Evolution in dimensionality reduction. In [111], an evolution visualization is proposed for discovered user groups (i.e., time-based visualization). The approach first discretizes time and normalizes user actions. Then it performs a PCA and returns a 2D projection for visualization. The approach in [112] goes one step further and highlights salient changes in each time bin.

4.1.4 Using **DV** for verifiability

It is often tedious for analysts to find the best parameters for the discovery optimization objective ρ through several trial-and-error sessions. For instance in discovery of common actions (Section 3.1.2), a low support often leads millions of groups, and a high support leads only a few groups (i.e., long-tail nature of group discovery [113]). Some **DV** approaches are proposed to assist analysts in parameter tuning [114], [115]. COQUITO [114] provides a visual interface which enables analysts to visually verify their desired parameters before rediscovery. An integrated discovery engine generates few approximated results with given parameters where the analyst verifies if they make sense.

4.2 Combining Discovery and Exploration

Group discovery often generates thousands to millions of user groups. This impedes analysts to verify and compare them one by one. Even a **DV** approach may be too cluttered on a large space of discovered groups. Combining discovery and exploration (denoted as **DE**) enables an interactive verification of discovered groups. For instance, Martin in Use Case 2 doesn't need to face all researcher groups at once, but in several iterations, where he gradually obtains his PC (Fig. 4).

A **DE** approach promotes a *discover-explore↻rediscover* paradigm, i.e., in each iteration, the analyst may decide to change parameters of the discovery objective ρ based on the new acquired knowledge and then request a rediscovery. **DE** has two following differences with discovery-alone (Section 3.1) and exploration-alone (Section 3.2) methods.

Feedback capturing. Each iteration in **DE** depends on previous iterations, which ensures that the exploration on discovered results is purposeful. This dependency is captured in the form of feedback, i.e., the analyst's preferences on explored groups.

Efficiency. **DE** is often described in an online context, where the analyst explores discovered groups, provides feedback, and receives new groups on-the-fly. While it is acceptable for a discovery method to run for hours offline, it is crucial for **DE** to have low latency between iterations [105], i.e., in sub-seconds.

In this section, first we review different formulations of feedback capturing in the literature. Then we discuss different types of **DE** approaches which enable an efficient feedback-based exploration on discovered groups either by focusing on user aggregations or groups. The main difference between the two **DE** categories is in their native support to consider user groups as first-class citizens. Table 4 summarizes the **DE** approaches.

4.2.1 Feedback formulation

An essential element of a **DE** approach is to let analysts provide feedback on explored results. **DE** then exploits this feedback in future iterations to customize the exploration and orient it towards the analyst's interests. Feedback is the core component of *interactive systems* which enforces HILDA aspects to the exploration process [18], [117]. Inspired from *progressive analytics* [118], feedback is often associated to a time limit to bound the analyst's waiting time in exploration. Time limit is usually set to $100ms$ to $500ms$ following continuity preserving latency [119]. A latency higher than $500ms$ has a negative impact on the analyst engagement towards the **DE** system.

Feedback can be seen as a *score* given by the analyst to all visited groups in the current iterations. These scores

TABLE 4
Summary of **DE** approaches

Type	Approach	Key advantage	Feedback model	Efficiency
User-based	IPM [69]	Simplicity of feedback capturing and updating analyst profile	Boolean	Random sampling
	IFE [35]	Convergence to a unique interestingness function	Partial order	Non-linear programming
	AIDE [116]	Intuitive decision tree model for analyst profile	Yes-no-maybe	Local optimizations
Group-based	DICE [13]	Dynamic exploration by forgetting previous choices	Boolean	Anticipation
	IUGA [18]	Discovering and exploring user groups as first-class citizens	Boolean	Greedy algorithm
	OCM [36]	Functionality with multiple group discovery algorithms	Yes-no-maybe	Greedy algorithm
	Cohana [32]	By-query exploration functionality for data scientists	Boolean	SQL optimizations

will collectively build an *analyst profile* (i.e., an aggregation of the analyst's interests in all exploration iterations so far) which enables **DE** to classify unexplored groups and identify interesting regions in user data. Feedback may be captured *explicitly* (i.e., the analyst professes her interest on groups) or *implicitly* (the system watches the analyst and derives her interests).

Explicit feedback. Following types of explicit feedback are proposed in the literature.

Boolean. Based on closed-world assumption [120], feedback is a boolean evaluation of each explored group, e.g., *like and dislike*, *preferred and not preferred* and *interesting and uninteresting* [69]. Boolean feedback is often fed into a binary classifier as hard constraints to find the interestingness class of unexplored groups. Due its simplicity, boolean feedback is the most adapted model in the literature [13], [18], [69].

Yes-no-maybe. Based on open-world assumption, feedback can also be expressed as *maybe interesting* [116]. This feedback is based on the intuition that no analyst has complete knowledge about what she is interested in. It is then cumbersome to limit her choices only to *interesting* and *uninteresting*. The third probable choice enables analysts to express a potential interest towards a group. **DE** interprets it as a soft constraint for classifying unexplored groups.

Integer score. The feedback can be an integer score from the range of a utility function, e.g., coverage, informativeness and representativeness [121], [122]. The advantage of this feedback is that a *total order* is captured on explored group. This order is employed to infer a global ordering over all groups. However, analysts often find it burdensome to provide a precise integer as feedback. For instance, it is not always straightforward for analysts to differentiate between a score 7 and 8 for a function with range $[0 - 10]$.

Partial order. To alleviate the curse of total orders, partial ordering is proposed where the analyst only needs to define which group is preferred to another group [123], [124], [125].

Implicit Feedback. Unlike explicit feedback where the analyst should clearly reflect her likes and dislikes, implicit feedback enables **DE** to capture *what the analyst may miss* instead of what the analyst has investigated before [126]. Following types of implicit feedback are proposed in the literature.

Gaze tracking. It is often the case that the analyst looks at some groups but simply forget to provide an explicit feedback. It shown in [127] that *gaze* has a strong correlation with *user attention*. Gaze can be captured by tracking eye movements (aka fixations) via eye trackers (webcams and augmented reality wearables) to derive interest for groups under the gaze area [128], [129].

Mouse tracking. Gaze tracking has privacy concerns [130]. An alternative option is mouse cursor tracking. It is shown in [131] that mouse gestures (e.g., click, hover, drag and drop) have a strong correlation with *user engagement*.

Session time. The time of investigation on a group is another indication of interest. The more the analyst focuses on a group, the more she is interested in that group [132].

4.2.2 Types of **DE** approaches

We recognize two types of **DE** approaches in the literature: *user-based* and *group-based*. In the former, the analyst profile is built upon single users and aggregated to form interests on groups. In the latter, the analyst profile is directly captured on groups. We review related works for each type and discuss their feedback model and efficiency considerations.

User-based DE. We discuss and compare the following approaches in the user-based category.

Interactive Pattern Mining (IPM) [69] employs a boolean feedback model, where groups are implemented as frequent patterns and support transactions are interpreted as group members. At each iteration, the system asks the analyst's feedback for k initially random groups as *interesting* or *uninteresting*. IPM iteratively updates a scoring function so that groups in future iterations align better with the analyst's interests. IPM forms the analyst profile based upon users. The scoring function maps each user to a non-negative real value, where higher values represent higher user interest. When the analyst shows interest on a group, the score of all its members will increase. Then an aggregation function computes a single score (called acceptance ratio) for each group as the multiplication of the members' score. Groups with high acceptance ratio will be explored in the next iteration. IPM employs random sampling to be efficient.

Interactive Feedback Exploration (IFE) [35] employs a partial ordering feedback model and groups are implemented as frequent patterns. IFE aims to predict parameters of an interestingness function which reflects the analyst's interests. At each iteration, k groups are sampled from the group space. Then the analyst provides her interest on the k groups by

expressing which groups are more interesting than others. The order is captured in form of a non-linear constraint which is then employed to refine the parameters of the interestingness function which is initialized at random. The system re-ranks the set of all groups according to the updated interestingness function. In the next iteration, k top-ranked groups will be explored. Akin to IPM, parameters are learned based upon users and then aggregated to obtain group-level scores. While IFE prioritizes effectiveness than efficiency (i.e., delivering better exploration results by sacrificing the execution time), it employs a non-linear programming model which is shown to be solvable within reasonable time [133].

Automated Interactive Data Exploration (AIDE) [116] employs a yes-no-maybe feedback model. Feedbacks are exploited in an active learning context to classify users into classes of *relevant* and *irrelevant*. AIDE distinguishes itself from other similar approaches by its sophisticated sampling approach. First, analyst feedback is received on a sample as *relevant*, *not relevant* and *maybe relevant*. A *relevant area* will then be formed in the space of users around the relevant sample. New samples are picked from the boundaries of this area. Feedback on the new sample will update the boundaries of the relevant area. In AIDE, the analyst profile is modeled as a *decision tree* which enables the verification of the analyst's tastes in an intuitive manner. To improve efficiency, AIDE employs local optimization tweaks in its sampling process such as avoiding the exploration of overlapping areas.

Group-based DE. We discuss and compare the following approaches in the group-based category.

Distributed Interactive Cube Exploration (DICE) [13] employs a boolean feedback model and groups are implemented as cubes. It builds a virtual lattice of user groups and lets the analyst perform by-example explorations on a group of interest in the lattice. DICE anticipates future exploration requests by exploiting the feedback received so far. Intuitively, if the analyst has already requested several upward explorations in the lattice (towards more general and larger groups), DICE infers that the analyst has an interest in rolling up in the user data to obtain a bigger picture. Hence groups in the upward direction will be explored first. DICE's anticipation engine contributes to efficiency by predicting the future workload and computing it in advance.

Interactive User Group Analysis (IUGA) [18] has a native support for user groups. IUGA employs a boolean feedback model. It maximizes a relevance function over the analyst profile. The relevance function consists of *diversity* (to provide different analysis directions) and *similarity* (to preserve the context of the analyst's interest). Diversity is measured as the amount of overlap (i.e., common users) between groups. Similarity is measured using Jaccard function. At each iteration, IUGA receives one group of interest and returns top-k relevant groups to explore. To improve efficiency, IUGA pre-computes the similarity between all pairs of groups, and exploits them in a greedy algorithm to maximize diversity.

OneClick Mining (OCM) [36] employs a yes-no-maybe feedback model and groups are implemented as patterns. OCM performs an online learning scheme to learn features that represent the analyst profile. At each iteration, the analyst provides feedback on each explored group as *keep* and *delete* which reflect her likes and dislikes, respectively. OCM follows closed-world assumption and mark other groups as *untouched*. OCM employs kept, deleted and untouched groups to update rules in the learning component. Updated rules are used to rank unseen groups by approximating their utility for further exploration. OCM is particularly designed for cases where more than one group discovery method contribute to the group space. At each iteration, the groups are sampled from the output of different methods based on a performance model built for each method using feedbacks. To improve efficiency, OCM employs a greedy approach for ranking groups at each iteration.

Cohort Query Processing (Cohana) [32] employs a boolean feedback model. It builds a cohort in the context of retention analysis[14] and lets analysts perform by-query explorations. Cohana limits the feedback to three following primitives which constitute a retention cohort: *birth selection* (i.e., when a cohort has been initiated), *age selection* (i.e., recency of a cohort) and *cohort aggregate*. A feedback on one of those primitives will return other cohorts with an updated value for that primitive.[15] Cohana identifies a unique SQL query for each cohort and performs some SQL optimizations to improve efficiency. Examples are data chunking (i.e. clustering users based on common birth time), RLE data compression (i.e., scanning less users) and two-level global dictionary encoding (i.e., skipping irrelevant users and groups).

4.2.3 Which DE approach to choose?

IPM is the method of choice for its *simplicity* of feedback capturing (using a boolean model) and updating the analyst feedback. In case *convergence* matters, learning-based methods are recommended (IFE, AIDE and OCM) to converge on one unique interestingness function. AIDE's particular interest is in its intuitive decision tree model for the analyst profile. OCM's particular interest is in its functionality with multiple discovery algorithms. While unlearning is a challenge for learning-based methods (as they accumulate analyst feedback in all iterations), local optimization approaches such as DICE, IUGA and Cohana can *forget previous choices* and enable a more dynamic exploration. Cohana and its by-query exploration mechanism appeals more interesting to data scientists while two others are example-based and can be easily picked by domain experts and information consumers.

4.3 Combining Exploration and Visualization

Traditionally, visualization methods are designed as one-shot and incapable of handling exploratory scenarios. Combining exploration and visualization (denoted as **EV**) enables interactive visualizations for user groups, aka visual analytics [142]. While **DV** approaches provide a big picture of groups, **EV** enables a visual navigation in the group space. **EV** finds its roots in the *visual information seeking mantra* of Ben Shneiderman, i.e., *overview first, zoom and filter, details on demand* [143].

14. *Retention https://mixpanel.com/retention/*

15. *A demo presentation of retention exploration is presented at https://youtu.be/XjPBynfUR8s*

TABLE 5
Summary of **EV** approaches

Type	Approach	Key advantage	Clear visualization
Exploration-based	Zenvisage [134]	By-query exploration functionality for data scientists	Restricted schema
	Data Tweening [135]	Tracking exploration transitions	Restricted mappings
	Vexus [136]	Exploring and visualizing user groups as first-class citizens	Few groups
	FlashView [137]	Intuitive fast visual exploration	Sampling
	ConVis [138]	By-facet exploration functionality for domain experts	Topic models
Visualization-based	Scented Widgets [139], [140]	Enabling wiser decision making in exploration	Limited attributes
	TextTile [76]	Exploration with OLAP-style operations	Summarization
	PTC [141]	Focusing on user actions	Layout & word-sizing
	Visualization Grammar [87]	Formalizing visual interactions	Customizability

Exploration and visualization aim for different goals. The goal of exploration is to guide analysts in the process of finding insights. The goal of visualization is to provide sensemaking for insights. These two components do not necessarily accompany each other at all times. For instance, NetLens [10] is a visualization-only approach, and IUGA [18] is an exploration-only approach. **EV** complements both exploration-alone and visualization-alone methods. Exploration-alone methods (Section 3.2) require a visualization layer to offer an intuitive visual interaction with analysts. Visualization-alone methods (Section 3.3) require exploration mechanisms to enable interactivity. Moreover, **EV** has two following differences with exploration-alone and visualization-alone methods.

Clear view. At each iteration of an **EV** approach, it is crucial to obtain a clear visualization of user groups with no confetti. Otherwise, the analyst is not able to provide proper feedback to proceed the exploration. It is shown in [4] that analysts are more comfortable in making decisions with less cluttered set of results.

Efficiency. Each iteration of an **EV** approach should be fast to preserve the train of thoughts of the analyst. This is extremely challenging due to the typical latency associated with visualization approaches.

In this section, we review different types of **EV** approaches and describe how they enable a visual exploration of groups by providing a clear visualization (Section 4.3.1), and then we discuss strategies to boost the efficiency of **EV** approaches (Section 4.3.3).

4.3.1 Types of *EV* approaches

We recognize two different types of **EV** approaches, *exploration-based* and *visualization-based*. In the former, a visualization approach is built on top of an existing exploration method. In the latter, an exploration mechanism is injected to an existing visualization method. Table 5 summarizes the **EV** approaches.

Exploration-based EV. The expressive power of visualization methods can be added to an existing exploration approach to enable visual inspection of explored groups and provide immediate insights. Visualized groups facilitate expressing partial needs for analysts, be it in form of query, facet, distribution, example, or evolution [134], [144]. We discuss and compare the following approaches in the exploration-based category.

Zenvisage. In [134], analysts interact with a visualization view using by-query exploration. Inspired from previous visualization query languages [145], [146], analyst's needs are expressed in terms of a SQL-like language called ZQL (Zenvisage Query Language). ZQL operates on top of a visual algebra which defines the logic behind the mapping function between user groups and visual variables. By defining a restrictive yet expressive schema for ZQL, Zenvisage guarantees that the visualizations are clear.

Data Tweening. In [135], a visualization view enables analysts to retain changes in exploration. It is often unclear for analysts how the exploration transitions from one iteration to another. Data Tweening visualizes additions, withdrawals and replacements of groups from one iteration to another, in an animated form. Data Tweening functions on a catalog of mapping functions, such as *order*, *rotate*, *nest* using which it visualizes the changes. The small size of the mapping catalog guarantees that the resulting visualization is clear.

Vexus. In [136], a visualization framework is proposed to provide native support for exploring user groups. The framework enables analysts to perform by-example explorations and observe results visually. The visualization helps analysts to have a better understanding of the explored groups and to observe more details about them. Analysts can seek to achieve either a single group in its entirety (e.g., finding an audience group for targeted advertisement), or identify several users of interest while exploring user groups (e.g., forming a conference PC). To prevent clutter, Vexus opts for visualizing only k diverse groups at each exploration iteration.

FlashView. In [137], a visualization interface is proposed for *fast exploration* of user groups. While there is often an offline pre-processing step for both visualization and exploration to prepare online execution, FlashView enables quick visual inspection of groups without pre-processing. FlashView employs approximate query processing (AQP) techniques to return the best possible exploration options without latency. The system builds some indexes on-the-go to deliver better results in future iterations. A clear visualization is guaranteed by sampling results.

ConVis. In [138], a visual by-facet exploration is proposed to analyze user commenting actions in social networks. The analyst can request a facet (e.g., ⟨gender, female⟩) to explore and visualize the sentiments of their comments. To reduce visual clutter, topic models are mined and visualized instead of comment's raw texts.

Visualization-based EV. The exploration layer added to an existing visualization method is often referred to as the *guidance layer* whose aim is to add directions to visual variables and ease group navigation. We discuss and compare the following approaches in the visualization-based category.

Scented Widgets. In [139], [140], an approximated distribution is visualized on top of each exploration option (in our case, group) to provide a sense of *how results will look like* in the next iteration. It guides analysts to make a wiser decision through their navigation. A clean view is guaranteed at each exploration iteration by limiting the number of investigated attributes in the visualization.

TextTile. In [76], the expressive power of OLAP-style exploration operations is added to visualization. TextTile implements groups as data cubes. In other words, TextTile enables a visualization-based OLAP. TextTile tailors a specific summarization layer for the OLAP operations which results in a clear and uncluttered visualization.

Parallel Tag Clouds (PTC). Word cloud is a visualization technique to represent the importance of a set of words using visual elements such as size and color [147]. In [141], a visual by-facet exploration approach is proposed to compare group actions together using word clouds (where each action is considered as a word). Each facet is visualized on one parallel coordinate. Then, user actions associated to that facet is visualized in form of word cloud. PTC considers both layout and word-sizing techniques to provide a clear visualization.

Visualization Grammar. There is a recent trend whose effort is to associate exploration methods to *visual grammars* [87], [148], [149]. Visual grammars facilitate creating, saving, and sharing visual analytics. Vega is among the most popular visual grammars in the literature [87]. *Signals* in Vega are capable of triggering exploration operations using JSON-like formalizations. Signals are dynamic variables that parameterize a visual element (e.g., a circle representing a group) for interactive behavior.[16] Also in [149], a grammar is introduced to build rules out of analysts' interactions and determine common visual patterns. Those rules are then employed to anticipate future visualization requests on groups. Full customizability of visual grammars enables analysts to operate on clear visualizations for their exploration tasks.

4.3.2 Which EV approach to choose?

A data scientist may be interested to employ Zenvisage and TextTile to have an exploration-wise control on the **EV** approach. She may also choose Visual Grammars to have the same control visualization-wise. With these methods, she can easily grasp intuitions behind exploration-visualization transforms and understand insights. She may want to have a more focused visual exploration. She focuses on group actions with PTC, on group changes with Data Tweening, and on transitions with Scented Widgets. However, in case *simplicity* has a priority, Vexus, FlashView and ConVis are the methods of choice. While FlashView builds visual explorations very fast, Vexus builds visualizations tailored for user groups. Also ConVis provides a by-facet exploration functionality which is preferred by domain experts.

16. VEGA *documentation on signals: https://vega.github.io/vega/docs/signals/*

4.3.3 Boosting *EV* approaches

To improve **EV** latencies, different indexing and sampling techniques are proposed in the literature. While we review the literature on boosting techniques for **EV**, it is important to notice that no efficiency improvement technique is specifically designed for group-based operations on user data.

Indexing. An index can be constructed on groups, their members and actions. Indexes enable quick retrieval of groups for fluid explorations and fast visualizations. Following index structures can be applied to an **EV** approach.

Nanocubes. In [150], [151] a cube-based indexing mechanism is proposed. By creating shared links between user attributes, Nanocubes renders visualizations faster than a typical screen refresh rate. The index is a trade-off between two state-of-the-art indexing mechanisms, i.e., Datavore [152] and ImMens [153]. The former is extremely fast but consumes lots of memory for its index structure. The latter optimizes memory usage at the price of slight latencies. Nanocubes is as efficient as ImMens while consuming less memory.

Cubrick. In [39], an in-memory DBMS is proposed to *serve group-based operations* as first-class citizens. Each group in Cubrick is called a *brick*. Common OLAP-style exploration operations can be performed quickly on bricks. Cubrick employs a row-major function to explore groups and uses a hashing map for a fast retrieval.

GraphVizDB. In [154], large graphs are indexed where nodes are users and edges are social links. GraphVizDB divides the whole graph into distinct regions (i.e., a set of groups) with respect to a Euclidean plane. Ideally, there is no overlap between regions and there are as few social links as possible between regions. Then a number of abstraction layers are constructed for each region. Once a group is requested by an **EV** approach, the index enables fast retrieval of groups thanks to the abstraction layers.

Sampling. Exploring and visualizing fewer groups improves the latency of **EV** approaches. Typically, random sampling considers equal probability for all groups to be explored. Smarter sampling mechanisms may be employed, such as *uniform* (i.e., pick random samples among data intervals, e.g., age categories) [155] and *stratified* (i.e., pick random samples from data clusters, e.g., different communities of user data) [13].

5 ALL-IN-ONE UGA

User group discovery, exploration and visualization address different aspects of UGA, separately or collectively. To benefit from their collective advantages, it is ideal to build an *all-in-one* UGA system (denoted **DEV**) where all

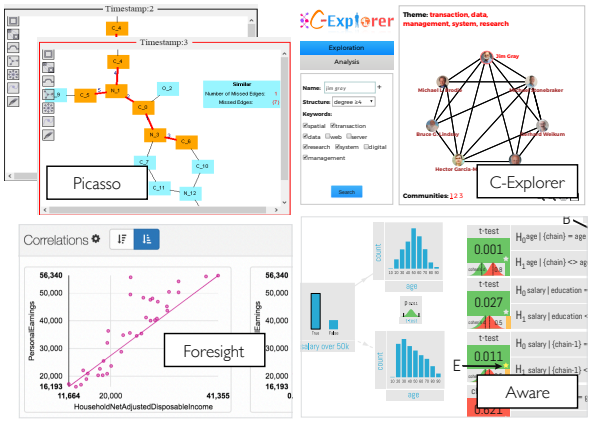


Fig. 5. **DEV** approaches.

those components are integrated. In other words, the aim is to build an automated *discover-explore-visualize* pipeline for user groups [156]. Note that the **DEV**'s literature is not yet established as it is an ongoing research topic. In this section, we first review the sparse literature on **DEV** approaches (Section 5.1) and then focus on two prominent open challenges towards a full-fledged **DEV** system: *connectivity between UGA components* (Section 5.3) and *evaluation* (Section 5.4).

5.1 Types of DEV approaches

A **DEV** approach consists of a layered architecture which covers all UGA components. It starts with discovering user groups, navigating in the group space using exploration, and finally visualizing results of interest. While these approaches offer the full UGA stack, they often miss connections between layers. Also it is often unclear how to evaluate the full UGA stack. We provide a summary of **DEV** approaches in Table 6 and discuss them as follows.

Progressive Connected Subgraph Substructure Search (Picasso). In [157], a user-friendly visual interface is proposed to explore user groups in form of graphs. Picasso first discovers attribute-based groups using *gSpan* graph mining algorithm [159]. Then it enables by-query explorations to navigate groups. It exploits Prague indexing mechanism [160] to make the rediscovery process faster: at each iteration, only a fragment of user groups that needs rediscovery will be mined again. Picasso visualizes groups in form of graphs (circles as users and lines as social links) using Java JDK. In Fig. 5, Picasso shows groups in form of graphs in two consecutive exploration iterations.

Community Explorer (C-Explorer). In [29], a web-based community analysis tool is proposed. While it offers a natural support for *attributed community search* as its discovery method [161] (i.e., finding communities as a function of both social links and user actions), it is generic enough to function with different discovery methods. In case more than one discovery method is functioning, a *comparison analysis* module is proposed to highlight differences in the results of each method. C-Explorer provides a by-facet exploration method. A visualization layer shows communities in form of graphs using JavaServer Pages (JSP). In Fig. 5, communities are mined on the DBLP bibliographical dataset. The name of a researcher (e.g., Jim Gray) is given as a facet.

Foresight. A recent trend in the **DEV**'s literature is to tighten the *discover-explore-visualize* loop, so that analysts can jump to visual insights directly [83], [162], [163]. In Foresight, visual insights are considered as first class citizens where analysts can instantly explore insights in a visual form [83]. An insight is defined as one or several groups with *outstanding statistical values* (e.g., a high linear correlation between a pair of attributes, the presence of extreme outliers for an attribute, etc.). Sketches, i.e., lossy representations of users data, is used to speed up insight discovery. Once insights are discovered, the analyst can either perform a by-distribution exploration to see other insights, or a by-example exploration to navigate other insightful groups. Foresight employs different visual views to visualize insights, e.g., scatter plots for linear relationships and box-and-whisker plots for outliers. In Fig. 5, a visualization view of Foresight depicts the regression of net adjusted disposal income and the personal earnings for a group in OECD dataset.

Aware. In [158], a statistical approach is proposed which validates whether a visual exploration on a discovered set of groups makes sense. This is extremely important to prevent analysts from making early (possibly incorrect) conclusions. Aware exploits PanoramicData visual interface [164] where analysts can perform by-facet explorations. Whenever a new facet is requested, Aware verifies and report False Discovery Rate (FDR) for the exploration. In Fig. 5, FDRs for a by-facet exploration by age is illustrated.

5.2 Which DEV approach to choose?

Discovery-wise, Picasso and Foresight focus on *group insights*, while C-Explorer and Aware focus on *group comparisons*. Exploration-wise, Picasso offers by-query exploration, C-Explorer and Aware offer by-facet exploration, and Foresight offers by-distribution and by-example explorations. Visualization-wise, Picasso and C-Explorer offer a traditional graph-style representation, while Foresight and Aware employ more sophisticated statistical-based visualizations.

5.3 Connectivity between UGA components

Most current **DEV** approaches assemble UGA components together without any *explicit connection* between them. The inter-connection is typically missing between UGA components due to the difference in their nature. For instance, although the exploration method is aware of the discovery method's output (i.e., the set $\mathcal{G}$), any other complementary information (such as size and distribution of user groups) is not necessarily communicated to the exploration method. A full-fledged **DEV** system should have a full inter-connection between its components.

Various *connectors* are proposed in the literature to make **DEV** components inter-connected. For instance, RODBC[17] and Tableau Data Engine[18] make a bridge between group

17. *RODBC Package for R: https://cran.r-project.org/web/packages/RODBC/RODBC.pdf*
18. *https://www.tableau.com/products/technology*

TABLE 6
Summary of **DEV** approaches

Approach	Functionality	Discovery	Exploration	Visualization
Picasso [157]	Group insights	gSpan	By-query	Graph-based
C-Explorer [29]	Group insights	Attributed community search	By-facet	Graph-based
Foresight [83]	Group comparison	Statistical computation	By-distribution and by-example	Statistical
Aware [158]	Group comparison	Statistical computation	By-facet	Distribution

discovery and the rest of the stack, i.e., exploration and visualization. However, most of these connectors have a limited usability. Making **DEV** components fully connected still remains a challenge. We make an example to elaborate on this challenge. Consider Melanie, a finance analyst, who wants to visualize groups of people who have made a deposit in a branch of her bank in NY state during Winter 2018. She discovers one million different groups and hands the group set to a visual exploration method. However, the visualization method has a limit of $100K$ points and introduces lags for more *(lack of discovery-visualization inter-connection)*. Moreover, the offered view has confetti. Melanie explores a subset of groups in Mount Pleasant to remove confetti. However, she receives no result as no user action exists in the requested city *(lack of exploration-visualization inter-connection)*. In case inter-connections were in place, Melanie could easily bypass the visualization limits using a sampling mechanism for exploration. Also a by-example exploration could be requested to prevent the empty answer problem by expanding the exploration to neighbors.

Although the problem of **DEV**'s inter-connections in its entirety is still unsolved, we review related work which covers this problem partially, and then discuss perspectives of full connectivity.

Lumira and VAS. In [165], [166], an inter-connection mechanism between exploration and visualization is proposed. Lumira is pixel-wise, i.e., explored groups will be pruned to match available pixels to visualize [166]. VAS is sampling-based, i.e., it picks a sampling rate that fits the visualization criteria defined by the analyst [165].

Ermac. In [167], the desiderata for a future data visualization management system (DVMS) is proposed. Ermac suggests to connect different UGA components together using a common declarative language and execute the whole UGA session with traditional database engines. This approach would ideally benefit from both *representativity* power of visualization and *performance* power of traditional DBMSs.

Full Connectivity. There should be a generic transformation function which facilitates the handshaking between **DEV** components. Such function translates the output of each component to the input of the consecutive component. The function should be *state-aware*, i.e., it should keep track of previous dialogs between the analyst and the system. It can then deliver necessary information to each component to make decisions, such as adding constraints to the discovery method and sampling explored groups. Such function can be developed using the state-of-the-art formalizations such as JSON documents in Vega [87] and declarative functions in DVMS [167].

The transformation function between discovery and exploration may observe *group space size* and *execution time* of the discovery method to pick a suitable sampling ratio for exploration. The function between exploration and visualization takes *exploration type* and *feedback* to pick the best uncluttered visualization views with relevant visual elements to the requested exploration. The function between visualization and discovery takes a *focused visual area* to manipulate the objective constraints of the discovery method.

5.4 Evaluation of UGA components

The evaluation of UGA components can be classified along three axes: *performance*, *quality*, and *user experience*. Discovery approaches are often evaluated using performance (e.g., response time, memory needs) and group quality. Visualization, on the other hand, is often measured using quality (e.g., accuracy) and user experience through user studies. Exploration evaluation is in-between and leverages all three axes. Combined approaches, i.e., **DE**, **DV** and **EV**, exploit a collection of evaluation measures. In this section, we review the related work on evaluation protocols for each UGA component. We then describe how a **DEV** system can be evaluated. Fig. 6 summarizes different evaluation protocols in the state of the art.

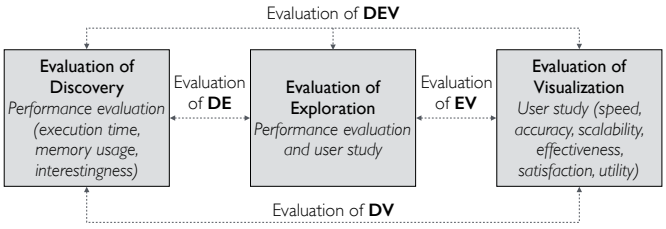


Fig. 6. UGA evaluation protocols.

5.4.1 Evaluation of each UGA components

We review various evaluation approaches proposed in the state-of-the-art for user group discovery, exploration and visualization.

Evaluation of group discovery. Analysts evaluate group discovery methods to find answers to questions such as "how much time does it take to discover groups?", "how much space is needed for the discovery?", "what is the quality of discovered groups?" Performance-based evaluation of group discovery heavily depends on the way the discovery objective ρ is computed. Quality-based evaluation, i.e., *to which extent groups in $\mathcal{G}$ are interesting for analysts?*, is performed using interestingness measures, which can be

TABLE 7
Interestingness measures for user groups [41], [42].

Measure	Intuition
Conciseness	Groups with shorter descriptions (fewer attribute values and actions) are more concise.
Coverage and Genericness	Groups which contain more users from $\mathcal{U}$ has a higher coverage (are more general).
Support	Larger groups (i.e., contain more users) have a higher support.
Reliability and Confidence	Groups are more reliable if their description hold for most users in $\mathcal{U}$.
Peculiarity and Surprisingness	Groups are more peculiar/surprising if their description deviates largely from other group descriptions.
Diversity and Entropy	Groups are more diverse if their description contains various attributes.
Novelty and Unexpectedness	Groups are novel if their description contradicts analyst's beliefs.
Usefulness	Groups are useful if they serve analyst's goals.

objective (i.e., as a function of the discovery objective ρ) or *subjective* (i.e., as a function of analyst's preferences). For instance, if $\rho = coverage$, then *support* can be considered as the objective interestingness function to evaluate groups. Additional measures such as *novelty* can be considered as the subjective interestingness function. Table 7 contains interestingness evaluation measures adapted for user groups. While performance and interestingness of discovered groups are evaluated, the size of the output (i.e., $|\mathcal{G}|$) is often missing in discovery evaluations.

Evaluation of group exploration. The aim of exploration evaluation is to verify the usefulness of the navigation mechanism (enforced by the exploration type θ) in enabling analysts to reach their goals [168]. Hence two different aspects should be evaluated: (i) if the exploration method succeeds in capturing analyst's profile, (ii) if the analyst's task is successfully fulfilled. These aspects can be evaluated quantitatively by comparing actual analyst's exploration steps with a *ground-truth*. In addition to quantitative measures, measuring human-oriented aspects in with a qualitative evaluation is also crucial.

Evaluation of the analyst's needs. We use quantitative measures for evaluating the analyst's needs at two granularities, i.e., local and global. While we need to understand how effective each exploration iteration is in satisfying the analyst's needs, we also need a comprehensive understanding of how all iterations together serve the analyst's goal. At a local level, *diversity*, *similarity* and *coverage* are used for evaluation [18], [61]. Diversity evaluates how well the method can explore different directions in user data. Similarity evaluates how well the method can preserve the context of an analyst's interest. Also coverage evaluates how well the exploration covers the whole user data. At the global level, we evaluate the *number of iterations* to reach a target. For instance, it is reported in [18] that for building a data management conference PC, around 12 iterations are needed.

Evaluation of task fulfillment. The satisfaction of analysts is a HILDA aspect of exploration which is evaluated qualitatively using pilot studies. Participants of a pilot study are often domain experts. A series of tasks are designed for experts to fulfill, e.g., *constructing a team of experts*, *detecting longitudinal factors of death in patient cohorts* and *finding a set of returning customers*. Task deployment for pilot studies is studied in [169], [170] where a taxonomy of ready-to-deploy tasks are provided. In a pilot study, each exploration iteration of the expert plus several measures of behavior such as *time-to-think* [171], will be recorded. Usually, an observer also notes her findings on her interactions with the expert during the pilot study.

Evaluation of group visualization. The usefulness of group visualization is evaluated in terms of quality and user experience via user studies [172], i.e., extended pilot studies whose participants are typical information consumers. Participants should answer questions about different aspects of the visualization. In [173], three levels of visualization evaluation (from system-centered to human-centered levels) are discussed:

Visualization view. A visualization tool may consist of several views. Each view (as a collection of visual elements) is evaluated separately. Participants answer questions about the *usefulness*, *informativeness* and *representativeness* of the view [93], [150], [174].

Visualization functionality. The collective behavior of the visualization suite (as a collection of its views) is evaluated at the functionality level. Participants should often follow simple atomic tasks such as ordering, picking and removing groups. Measures such as *effectiveness*, *satisfaction* and *time to insight* are computed for performed tasks to evaluate the visualization functionality [76], [82], [103].

Visualization interaction. The interaction between analysts and the visualization suite is evaluated at this level. Participants perform more complex tasks to find out *how well the approach can assist analysts in achieving their targets*. Measures such as *adoption*, *productivity*, *utility* and *learnability* are computed for performed tasks to evaluate the visualization interactivity [18], [118].

5.4.2 Evaluation of DEV

The need for a principled evaluation methodology arises when designing a **DEV** approach comprising all UGA components. Despite the established body of related work for evaluating discovery, exploration and visualization, there is no evaluation mechanism for **DEV** in its entirety. A valid question is whether *we can evaluate **DEV** with a combination of methods proposed to evaluate its components?* Adapting discovery-based evaluations (i.e., performance and quality axes) do not provide perspective on analyst-based aspects. Adapting visualization-based evaluation protocols

(i.e., quality and user experience axes) to **DEV** does not cover its quantitative aspects (e.g., how fast its discovery method performs.) On the other hand, user studies may be biased and incomplete [175]. We discuss four novel opportunities of **DEV** evaluation, as follows.

Isolation. The most popular approach is to isolate human-oriented UGA components and evaluate remaining components using traditional discovery-based measures, such as execution time and memory needs. For human-oriented components, a user study is designed. Although isolation enables a thorough evaluation on all components of a **DEV** approach, it suffers from two drawbacks. First, in all UGA components, the boundaries of human-oriented and system-oriented aspects are fuzzy. For instance, exploration is fired by an analyst, but some system-oriented aspects (e.g., coverage) are also associated to exploration. Second, isolated evaluation assesses each single **DEV** component, but not inter-communications between those components.

Crowdsourcing. Crowdsourcing platforms such as Amazon Mechanical Turk[19], Crowd4U[20] and CrowdFlower[21] scale up user studies by providing access to a large audience of information consumers [176]. The high confidence associated to a user study with a large population dissolves doubts on bias and incompletion. It is shown in [177] that for a dataset with $|\mathcal{U}| \geq 100K$, at least 1100 participants are needed to achieve results with an error margin of $\pm 3\%$.

Quantified user study. User studies can be enriched with quantified measures to complement participants' answers. While responding to questions, measures such as *time-to-think*, *mouse actions*, *eye movements*, *scrolling actions*, *dragging speed*, *number of backtrackings*, *number of cycles* and *number of restarts* will be recorded for participants [178]. This enables both qualitative and quantitative evaluation of **DEV**.

Benchmarking. The quality of a **DEV** approach can be assessed by comparing it against standard tests, i.e., benchmarks. Benchmarks are a common practice in the database community (e.g., Oracle TPC benchmark [179], LDBC Social Network Benchmark [180] and REACT data exploration benchmark [181]). IDEBench is proposed in [182], [183], [184] as an **EV** benchmark. The benchmark contains common exploration primitives (such as aggregation and filtering) which are empirically observed in a range of user studies. This benchmark impacts the way an **EV** approach is evaluated, as both aspects of exploration and visualization are captured. A **DEV** benchmark, however, should consist of analyst traces, i.e., a recorded session (using screen captures, recorded voice, I/O capture, etc.) of analyst actions in all UGA components, i.e., discovery, exploration and visualization.

In summary, the HILDA aspect of UGA is a new axis in evaluating a **DEV** system which goes far beyond typical quantitative measures and hence needs further research. For this aim, human factors (e.g., motivation and satisfaction) in user group exploration and their influence on the **DEV** outcome should be studied either using user studies and crowdsourcing platforms, or by comparing against a gold standard, i.e., a benchmark.

19. *https://www.mturk.com*
20. *https://crowd4u.org*
21. *https://www.crowdflower.com*

6 Conclusion

In this survey, we motivate group-level analysis of user data, i.e., User Group Analytics (UGA). Groups enable new insights and address peculiarities of user data such as noise and sparsity. We discuss the usability of UGA for different roles of users, i.e., data scientists, domain experts and information consumers. We discuss related work for three principled components of UGA, i.e., discovery, exploration and visualization. To benefit from their collective advantages, we review work which combines UGA components together. We discuss research opportunities and challenges of designing an all-in-one UGA system. Last, we discuss evaluation opportunities of each single UGA component as well as their combination.

Acknowledgment

This work is supported by CDP LIFE project under grant C7H-ID16-PR4-LIFELIG.

References

[1] B. Omidvar-Tehrani and S. Amer-Yahia, "User group analytics: Discovery, exploration and visualization," in *CIKM*. ACM, 2018.
[2] ——, "Data pipelines for user group analytics," in *SIGMOD*. ACM, 2019.
[3] L. Cao, "Behavior informatics to discover behavior insight for active and tailored client management," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2017, pp. 15–16.
[4] C. C. Gramazio, K. B. Schloss, and D. H. Laidlaw, "The relation between visualization size, grouping, and user performance," *IEEE transactions on visualization and computer graphics*, vol. 20, no. 12, pp. 1953–1962, 2014.
[5] J. Doodson, J. Gavin, and R. Joiner, "Getting acquainted with groups and individuals: Information seeking, social uncertainty and social network sites." in *ICWSM*, 2013.
[6] F. Du, C. Plaisant, N. Spring, and B. Shneiderman, "Finding similar people to guide life choices: Challenge, design, and evaluation," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 2017, pp. 5498–5544.
[7] A. Mangalampalli, A. Ratnaparkhi, A. O. Hatch, A. Bagherjeiran, R. Parekh, and V. Pudi, "A feature-pair-based associative classification approach to look-alike modeling for conversion-oriented user-targeting in tail campaigns," in *WWW*, 2011, pp. 85–86.
[8] T. Rekatsinas, X. Chu, I. F. Ilyas, and C. Ré, "Holoclean: Holistic data repairs with probabilistic inference," *arXiv preprint arXiv:1702.00820*, 2017.
[9] G. Miller, "Human memory and the storage of information," *IRE Transactions on Information Theory*, vol. 2, no. 3, pp. 129–137, 1956.
[10] H. Kang, C. Plaisant, B. Lee, and B. B. Bederson, "Netlens: iterative exploration of content-actor network data," *Information Visualization*, vol. 6, no. 1, pp. 18–31, 2007.
[11] N. Henry, J.-D. Fekete, and M. J. McGuffin, "Nodetrix: a hybrid visualization of social networks," *IEEE transactions on visualization and computer graphics*, vol. 13, no. 6, pp. 1302–1309, 2007.
[12] B. C. Kwon, S.-H. Kim, S. Lee, J. Choo, J. Huh, and J. S. Yi, "Visohc: Designing visual analytics for online health communities," *IEEE transactions on visualization and computer graphics*, vol. 22, no. 1, pp. 71–80, 2016.
[13] N. Kamat, P. Jayachandran, K. Tunga, and A. Nandi, "Distributed and interactive cube exploration," in *Data Engineering (ICDE), 2014 IEEE 30th International Conference on*. IEEE, 2014, pp. 472–483.
[14] S. Roy and D. Suciu, "A formal approach to finding explanations for database queries," in *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*. ACM, 2014, pp. 1579–1590.

[15] H. Zhang, Y. Diao, and A. Meliou, "Exstream: Explaining anomalies in event stream monitoring." in *EDBT*, 2017, pp. 156–167.

[16] E. Wu and S. Madden, "Scorpion: Explaining away outliers in aggregate queries," *Proceedings of the VLDB Endowment*, vol. 6, no. 8, pp. 553–564, 2013.

[17] S. Roy, L. Orr, and D. Suciu, "Explaining query answers with explanation-ready databases," *Proceedings of the VLDB Endowment*, vol. 9, no. 4, pp. 348–359, 2015.

[18] B. Omidvar-Tehrani, S. Amer-Yahia, and A. Termier, "Interactive user group analysis," in *CIKM*. ACM, 2015, pp. 403–412.

[19] S. Sidana, S. Mishra, S. Amer-Yahia, M. Clausel, and M.-R. Amini, "Health monitoring on social media over time," in *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 2016, pp. 849–852.

[20] M. Kirchgessner, V. Leroy, A. Termier, S. Amer-Yahia, and M.-C. Rousset, "Toppi: An efficient algorithm for item-centric mining," in *International Conference on Big Data Analytics and Knowledge Discovery*. Springer, 2016, pp. 19–33.

[21] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 5, no. 4, p. 19, 2016.

[22] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere, "The million song dataset," in *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*, 2011.

[23] M. Monroe, R. Lan, H. Lee, C. Plaisant, and B. Shneiderman, "Temporal event sequence simplification," *IEEE transactions on visualization and computer graphics*, vol. 19, no. 12, pp. 2227–2236, 2013.

[24] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen, "Improving recommendation lists through topic diversification," in *Proceedings of the 14th international conference on World Wide Web*. ACM, 2005, pp. 22–32.

[25] B. Golshan, T. Lappas, and E. Terzi, "Sofia search: a tool for automating related-work search," in *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*. ACM, 2012, pp. 621–624.

[26] S. Amer-Yahia, B. O. Tehrani, S. B. Roy, and N. Shabib, "Group recommendation with temporal affinities." in *EDBT*, 2015, pp. 421–432.

[27] H. Zhong, C. Liu, X. Lu, and H. Xiong, "To be or not to be friends: Exploiting social ties for venture investments," in *Data Mining (ICDM), 2016 IEEE 16th International Conference on*. IEEE, 2016, pp. 699–708.

[28] M. E. Newman, "Detecting community structure in networks," *The European Physical Journal B-Condensed Matter and Complex Systems*, vol. 38, no. 2, pp. 321–330, 2004.

[29] Y. Fang, R. Cheng, S. Luo, J. Hu, and K. Huang, "C-explorer: Browsing communities in large graphs," in *Proc. VLDB Endowment*, vol. 10, 2017.

[30] A. Goyal, F. Bonchi, and L. V. Lakshmanan, "Discovering leaders from community actions," in *Proceedings of the 17th ACM conference on Information and knowledge management*. ACM, 2008, pp. 499–508.

[31] C. Bron and J. Kerbosch, "Algorithm 457: finding all cliques of an undirected graph," *Communications of the ACM*, vol. 16, no. 9, pp. 575–577, 1973.

[32] D. Jiang, Q. Cai, G. Chen, H. Jagadish, B. C. Ooi, K.-L. Tan, and A. K. Tung, "Cohort query processing," *Proceedings of the VLDB Endowment*, vol. 10, no. 1, pp. 1–12, 2016.

[33] A. G. Nikolaev, S. Gore, and V. Govindaraju, "Engagement capacity and engaging team formation for reach maximization of online social media platforms." in *KDD*, 2016, pp. 225–234.

[34] S. Amer-Yahia, S. Kleisarchaki, N. K. Kolloju, L. V. Lakshmanan, and R. H. Zamar, "Exploring rated datasets with rating maps," in *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2017, pp. 1411–1419.

[35] D. Xin, X. Shen, Q. Mei, and J. Han, "Discovering interesting patterns through user's interactive feedback," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2006, pp. 773–778.

[36] M. Boley, M. Mampaey, B. Kang, P. Tokmakov, and S. Wrobel, "One click mining: Interactive local pattern discovery through implicit preference and performance learning," in *Proceedings of the ACM SIGKDD Workshop on Interactive Data Exploration and Analytics*. ACM, 2013, pp. 27–35.

[37] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, *Automatic subspace clustering of high dimensional data for data mining applications*. ACM, 1998, vol. 27.

[38] R. Srikant and R. Agrawal, "Mining generalized association rules," *ACM*, 1995.

[39] P. Pedreira, C. Croswhite, and L. Bona, "Cubrick: indexing millions of records per second for interactive analytics," *Proceedings of the VLDB Endowment*, vol. 9, no. 13, pp. 1305–1316, 2016.

[40] B. Omidvar-Tehrani, S. Amer-Yahia, E. Simon, F. C. Zegarra, J. L. D. Comba, and V. Moreira, "Userdev: A mixed-initiative system for user group analytics," in *HILDA@SIGMOD*. ACM, 2019.

[41] L. Geng and H. J. Hamilton, "Interestingness measures for data mining: A survey," *ACM Computing Surveys (CSUR)*, vol. 38, no. 3, p. 9, 2006.

[42] M. Kirchgessner, V. Leroy, S. Amer-Yahia, and S. Mishra, "Testing interestingness measures in practice: A large-scale analysis of buying patterns," in *Data Science and Advanced Analytics (DSAA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 547–556.

[43] M. Das, S. Amer-Yahia, G. Das, and C. Yu, "Mri: Meaningful interpretations of collaborative ratings," *Proceedings of the VLDB Endowment*, vol. 4, no. 11, pp. 1063–1074, 2011.

[44] B. Omidvar-Tehrani, S. Amer-Yahia, P.-F. Dutot, and D. Trystram, "Multi-objective group discovery on the social web," in *ECML/PKDD*. Springer, 2016, pp. 296–312.

[45] J. Liu, L. Xiong, J. Pei, J. Luo, and H. Zhang, "Finding pareto optimal groups: group-based skyline," *Proceedings of the VLDB Endowment*, vol. 8, no. 13, pp. 2086–2097, 2015.

[46] I. M. Baytas, C. Xiao, X. Zhang, F. Wang, A. K. Jain, and J. Zhou, "Patient subtyping via time-aware lstm networks," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2017, pp. 65–74.

[47] M. Plantié and M. Crampes, "Survey on social community detection," in *Social media retrieval*. Springer, 2013, pp. 65–85.

[48] J. Kim and J.-G. Lee, "Community detection in multi-layer graphs: A survey," *ACM SIGMOD Record*, vol. 44, no. 3, pp. 37–48, 2015.

[49] G. Rossetti and R. Cazabet, "Community discovery in dynamic networks: a survey," *ACM Computing Surveys (CSUR)*, vol. 51, no. 2, p. 35, 2018.

[50] S. Harenberg, G. Bello, L. Gjeltema, S. Ranshous, J. Harlalka, R. Seay, K. Padmanabhan, and N. Samatova, "Community detection in large-scale networks: a survey and empirical evaluation," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 6, no. 6, pp. 426–439, 2014.

[51] J. Leskovec, K. J. Lang, and M. Mahoney, "Empirical comparison of algorithms for network community detection," in *Proceedings of the 19th international conference on World wide web*. ACM, 2010, pp. 631–640.

[52] M. L. Gregory, D. W. Engel, E. B. Bell, A. Piatt, S. Dowson, and A. J. Cowell, "Automatically identifying groups based on content and collective behavioral patterns of group members." in *ICWSM*, 2011.

[53] A. E. Sariyüce and A. Pinar, "Fast hierarchy construction for dense subgraphs," *Proceedings of the VLDB Endowment*, vol. 10, no. 3, pp. 97–108, 2016.

[54] C.-Y. Shen, L.-H. Huang, D.-N. Yang, H.-H. Shuai, W.-C. Lee, and M.-S. Chen, "On finding socially tenuous groups for online social networks," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2017, pp. 415–424.

[55] L. Soulier, L. Tamine, and G.-H. Nguyen, "Answering twitter questions: a model for recommending answerers through social collaboration," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 2016, pp. 267–276.

[56] S. Yu, E. Christakopoulou, and A. Gupta, "Identifying decision makers from professional social networks," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 333–342.

[57] D.-N. Yang, Y.-L. Chen, W.-C. Lee, and M.-S. Chen, "On social-temporal group query with acquaintance constraint," *Proceedings of the VLDB Endowment*, vol. 4, no. 6, pp. 397–408, 2011.

[58] P. Lee, L. V. Lakshmanan, and E. E. Milios, "Incremental cluster evolution tracking from highly dynamic network data," in *Data Engineering (ICDE), 2014 IEEE 30th International Conference on*. IEEE, 2014, pp. 3–14.

[59] J. Yang, J. McAuley, J. Leskovec, P. LePendu, and N. Shah, "Finding progression stages in time-evolving event sequences," in *Proceedings of the 23rd international conference on World wide web*. ACM, 2014, pp. 783–794.

[60] T. Zhang, P. Cui, C. Faloutsos, Y. Lu, H. Ye, W. Zhu, and S. Yang, "Come-and-go patterns of group evolution: A dynamic model," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 1355–1364.

[61] Y. Yang, D. Yan, H. Wu, J. Cheng, S. Zhou, and J. C. Lui, "Diversified temporal subgraph pattern mining." in *KDD*, 2016, pp. 1965–1974.

[62] F. Lemmerich, M. Becker, P. Singer, D. Helic, A. Hotho, and M. Strohmaier, "Mining subgroups with exceptional transition behavior," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 965–974.

[63] M. Liu, Z. Liu, C. Zhang, K. Zhang, Q. Yuan, T. Hanratty, and J. Han, "Urbanity: A system for interactive exploration of urban dynamics from streaming human sensing data," *Dim*, vol. 1501, p. 2, 2017.

[64] R. West and J. Leskovec, "Automatic versus human navigation in information networks." in *ICWSM*, 2012.

[65] Y. Li, J. Fan, D. Zhang, and K.-L. Tan, "Discovering your selling points: Personalized social influential tags exploration," in *Proceedings of the 2017 ACM International Conference on Management of Data*. ACM, 2017, pp. 619–634.

[66] F. Bonchi, F. Giannotti, C. Lucchese, S. Orlando, R. Perego, and R. Trasarti, "Conquest: a constraint-based querying system for exploratory pattern discovery," in *Data Engineering, 2006. ICDE'06. Proceedings of the 22nd International Conference on*. IEEE, 2006, pp. 159–159.

[67] F. Bonchi, F. Giannotti, A. Mazzanti, and D. Pedreschi, "Exante: Anticipated data reduction in constrained pattern mining," in *PKDD*, vol. 2838. Springer, 2003, pp. 59–70.

[68] A. Balmin, L. Colby, E. Curtmola, Q. Li, F. Özcan, S. Srinivas, and Z. Vagena, "Seda: a system for search, exploration, discovery, and analysis of xml data," *Proceedings of the VLDB Endowment*, vol. 1, no. 2, pp. 1408–1411, 2008.

[69] M. Bhuiyan, S. Mukhopadhyay, and M. A. Hasan, "Interactive pattern mining on hidden data: a sampling-based solution," in *Proceedings of the 21st ACM international conference on Information and knowledge management*. ACM, 2012, pp. 95–104.

[70] K. Feng, G. Cong, S. S. Bhowmick, W.-C. Peng, and C. Miao, "Towards best region search for data exploration," in *Proceedings of the 2016 International Conference on Management of Data*. ACM, 2016, pp. 1055–1070.

[71] M. Sozio and A. Gionis, "The community-search problem and how to plan a successful cocktail party," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2010, pp. 939–948.

[72] Y. Fang, R. Cheng, X. Li, S. Luo, and J. Hu, "Effective community search over large spatial graphs," *Proceedings of the VLDB Endowment*, vol. 10, no. 6, pp. 709–720, 2017.

[73] Y. Fang, Z. Wang, R. Cheng, H. Wang, and J. Hu, "Effective and efficient community search over large directed graphs," *IEEE Transactions on Knowledge and Data Engineering*, 2018.

[74] N. Yan, C. Li, S. B. Roy, R. Ramegowda, and G. Das, "Facetedpedia: enabling query-dependent faceted search for wikipedia," in *Proceedings of the 19th ACM international conference on Information and knowledge management*. ACM, 2010, pp. 1927–1928.

[75] A. R. Khan and H. Garcia-Molina, "Crowddqs: Dynamic question selection in crowdsourcing systems," in *Proceedings of the 2017 ACM International Conference on Management of Data*. ACM, 2017, pp. 1447–1462.

[76] C. Felix, A. V. Pandey, and E. Bertini, "Texttile: an interactive visualization tool for seamless exploratory analysis of structured data and unstructured text," *IEEE transactions on visualization and computer graphics*, vol. 23, no. 1, pp. 161–170, 2017.

[77] S. Amer-Yahia, S. Kleisarchaki, N. K. Kolloju, L. V. Lakshmanan, and R. H. Zamar, "Exploring rated datasets with rating maps," in *WWW*, 2017.

[78] B. Omidvar-Tehrani and S. Amer-Yahia, "Online lattice-based abstraction of user groups," in *International Conference on Database and Expert Systems Applications*. Springer, 2017, pp. 95–110.

[79] E. Galbrun and P. Miettinen, "Interactive redescription mining," in *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*. ACM, 2014, pp. 1079–1082.

[80] M. Kahng, S. B. Navathe, J. T. Stasko, and D. H. P. Chau, "Interactive browsing and navigation in relational databases," *Proceedings of the VLDB Endowment*, vol. 9, no. 12, pp. 1017–1028, 2016.

[81] D. Mottin, M. Lissandrini, Y. Velegrakis, and T. Palpanas, "New trends on exploratory methods for data analytics," *Proceedings of the VLDB Endowment*, vol. 10, no. 12, pp. 1977–1980, 2017.

[82] D. Gotz and H. Stavropoulos, "Decisionflow: Visual analytics for high-dimensional temporal event sequence data," *IEEE transactions on visualization and computer graphics*, vol. 20, no. 12, pp. 1783–1792, 2014.

[83] Ç. Demiralp, P. J. Haas, S. Parthasarathy, and T. Pedapati, "Foresight: recommending visual insights," *Proceedings of the VLDB Endowment*, vol. 10, no. 12, pp. 1937–1940, 2017.

[84] S. Idreos, O. Papaemmanouil, and S. Chaudhuri, "Overview of data exploration techniques," in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. ACM, 2015, pp. 277–281.

[85] J. Heer and J. M. Hellerstein, "Tutorial on data visualization and social data analysis," *Proceedings of the VLDB Endowment*, vol. 2, no. 2, pp. 1656–1657, 2009.

[86] A. V. Pandey, A. Manivannan, O. Nov, M. Satterthwaite, and E. Bertini, "The persuasive power of data visualization," *IEEE transactions on visualization and computer graphics*, vol. 20, no. 12, pp. 2211–2220, 2014.

[87] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer, "Vega-lite: A grammar of interactive graphics," *IEEE transactions on visualization and computer graphics*, vol. 23, no. 1, pp. 341–350, 2017.

[88] G. D. Battista, P. Eades, R. Tamassia, and I. G. Tollis, *Graph drawing: algorithms for the visualization of graphs*. Prentice Hall PTR, 1998.

[89] I. Herman, G. Melançon, and M. S. Marshall, "Graph visualization and navigation in information visualization: A survey," *IEEE Transactions on visualization and computer graphics*, vol. 6, no. 1, pp. 24–43, 2000.

[90] D. Koutra, D. Jin, Y. Ning, and C. Faloutsos, "Perseus: an interactive large-scale graph mining and visualization tool," *Proceedings of the VLDB Endowment*, vol. 8, no. 12, pp. 1924–1927, 2015.

[91] Z. Liu, B. Kerr, M. Dontcheva, J. Grover, M. Hoffman, and A. Wilson, "Coreflow: Extracting and visualizing branching patterns from event sequences," in *Computer Graphics Forum*, vol. 36, no. 3. Wiley Online Library, 2017, pp. 527–538.

[92] G. N. Oliveira, J. L. Sotomayor, R. P. Torchelsen, C. T. Silva, and J. L. Comba, "Visual analysis of bike-sharing systems," *Computers & Graphics*, vol. 60, pp. 119–129, 2016.

[93] V. Machado, R. Leite, F. Moura, S. Cunha, F. Sadlo, and J. L. Comba, "Visual soccer match analysis using spatiotemporal positions of players," *Computers & Graphics*, vol. 68, pp. 84–95, 2017.

[94] T. Polk, J. Yang, Y. Hu, and Y. Zhao, "Tennivis: Visualization for tennis match analysis," *IEEE transactions on visualization and computer graphics*, vol. 20, no. 12, pp. 2339–2348, 2014.

[95] J. Wei, Z. Shen, N. Sundaresan, and K.-L. Ma, "Visual cluster exploration of web clickstream data," in *Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on*. IEEE, 2012, pp. 3–12.

[96] J. Stasko and E. Zhang, "Focus+ context display and navigation techniques for enhancing radial, space-filling hierarchy visualizations," in *Information Visualization, 2000. InfoVis 2000. IEEE Symposium on*. IEEE, 2000, pp. 57–65.

[97] P. Klemm, K. Lawonn, S. Glaßer, U. Niemann, K. Hegenscheid, H. Völzke, and B. Preim, "3d regression heat map analysis of population study data," *IEEE transactions on visualization and computer graphics*, vol. 22, no. 1, pp. 81–90, 2016.

[98] A. Makanju, S. Brooks, A. N. Zincir-Heywood, and E. E. Milios, "Logview: Visualizing event log clusters," in *Privacy, Security and Trust, 2008. PST'08. Sixth Annual Conference on*. IEEE, 2008, pp. 99–108.

[99] J. B. Kruskal and J. M. Landwehr, "Icicle plots: Better displays for hierarchical clustering," *The American Statistician*, vol. 37, no. 2, pp. 162–168, 1983.

[100] J. Zhao, C. Collins, F. Chevalier, and R. Balakrishnan, "Interactive exploration of implicit and explicit relations in faceted datasets," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2080–2089, 2013.

[101] G. Wang, X. Zhang, S. Tang, H. Zheng, and B. Y. Zhao, "Unsupervised clickstream clustering for user behavior analysis,"

in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 2016, pp. 225–236.

[102] I. Cadez, D. Heckerman, C. Meek, P. Smyth, and S. White, "Model-based clustering and visualization of navigation patterns on a web site," *Data Mining and Knowledge Discovery*, vol. 7, no. 4, pp. 399–424, 2003.

[103] Y. Chen, P. Xu, and L. Ren, "Sequence synopsis: Optimize visual summary of temporal event data," *IEEE Transactions on Visualization and Computer Graphics*, 2017.

[104] M. H. Shimabukuro, E. F. Flores, and F. Maria Cristina, "de oliveira, and haim levkowitz,"," in *Coordinated Views to Assist Exploration of Spatio-Temporal Data: A Case Study," 2nd International Conference on Coordinated & Multiple Views in Exploratory Visualization (CMV'04)*, 2004, pp. 107–117.

[105] J.-D. Fekete and C. Plaisant, "Interactive information visualization of a million items," in *Information Visualization, 2002. INFOVIS 2002. IEEE Symposium on*. IEEE, 2002, pp. 117–124.

[106] I. T. Jolliffe, "Mathematical and statistical properties of population principal components," *Principal Component Analysis*, pp. 10–28, 2002.

[107] I. Borg and P. J. Groenen, *Modern multidimensional scaling: Theory and applications*. Springer Science & Business Media, 2005.

[108] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[109] T. Elgamal, M. Yabandeh, A. Aboulnaga, W. Mustafa, and M. Hefeeda, "spca: Scalable principal component analysis for big data on distributed platforms," in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. ACM, 2015, pp. 79–91.

[110] H. Xu, Z. Li, S. Guo, and K. Chen, "Cloudvista: interactive and economical visual cluster analysis for big data in the cloud," *Proceedings of the VLDB Endowment*, vol. 5, no. 12, pp. 1886–1889, 2012.

[111] S. van den Elzen, D. Holten, J. Blaas, and J. J. van Wijk, "Reducing snapshots to points: A visual analytics approach to dynamic network exploration," *IEEE transactions on visualization and computer graphics*, vol. 22, no. 1, pp. 1–10, 2016.

[112] J. Abello, S. Hadlak, H. Schumann, and H.-J. Schulz, "A modular degree-of-interest specification for the visual analysis of large dynamic networks," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 3, pp. 337–350, 2014.

[113] S. Goel, A. Broder, E. Gabrilovich, and B. Pang, "Anatomy of the long tail: ordinary people with extraordinary tastes," in *Proceedings of the third ACM international conference on Web search and data mining*. ACM, 2010, pp. 201–210.

[114] J. Krause, A. Perer, and H. Stavropoulos, "Supporting iterative cohort construction with visual temporal queries," *IEEE transactions on visualization and computer graphics*, vol. 22, no. 1, pp. 91–100, 2016.

[115] P. Klemm, S. Oeltze-Jafra, K. Lawonn, K. Hegenscheid, H. Völzke, and B. Preim, "Interactive visual analysis of image-centric cohort study data," *IEEE transactions on visualization and computer graphics*, vol. 20, no. 12, pp. 1673–1682, 2014.

[116] K. Dimitriadou, O. Papaemmanouil, and Y. Diao, "Aide: an active learning-based approach for interactive data exploration," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 11, pp. 2842–2856, 2016.

[117] A. Nandi and H. Jagadish, "Guided interaction: Rethinking the query-result paradigm," *Proceedings of the VLDB Endowment*, vol. 4, no. 12, pp. 1466–1469, 2011.

[118] E. Zgraggen, A. Galakatos, A. Crotty, J.-D. Fekete, and T. Kraska, "How progressive visualizations affect exploratory analysis," *IEEE transactions on visualization and computer graphics*, 2016.

[119] Z. Liu and J. Heer, "The effects of interactive latency on exploratory visual analysis," *IEEE transactions on visualization and computer graphics*, vol. 20, no. 12, pp. 2122–2131, 2014.

[120] J. Minker, "On indefinite databases and the closed world assumption," in *International Conference on Automated Deduction*. Springer, 1982, pp. 292–308.

[121] N. Hariri, B. Mobasher, and R. Burke, "Adapting to user preference changes in interactive recommendation." in *IJCAI*, vol. 15, 2015, pp. 4268–4274.

[122] G. Adomavicius and A. Tuzhilin, "Context-aware recommender systems," in *Recommender systems handbook*. Springer, 2015, pp. 191–226.

[123] K. Panev and S. Michel, "Reverse engineering top-k database queries with paleo." in *EDBT*, 2016, pp. 113–124.

[124] R. Bespinyowong, W. Chen, H. Jagadish, and Y. Ma, "Exrank: an exploratory ranking interface," *Proceedings of the VLDB Endowment*, vol. 9, no. 13, pp. 1529–1532, 2016.

[125] R. C.-W. Wong, A. W.-C. Fu, J. Pei, Y. S. Ho, T. Wong, and Y. Liu, "Efficient skyline querying with variable user preferences on nominal attributes," *Proceedings of the VLDB Endowment*, vol. 1, no. 1, pp. 1032–1043, 2008.

[126] B. Omidvar-Tehrani, P. Souza Neto, P. Felipe, and F. Bento, "Geoguide: An interactive guidance approach for spatial data," in *DARLI-AP*. IEEE, 2017.

[127] M. H. Fischer, "An investigation of attention allocation during sequential eye movement tasks," *The Quarterly Journal of Experimental Psychology: Section A*, vol. 52, no. 3, pp. 649–677, 1999.

[128] R. J. Jacob and K. S. Karn, "Eye tracking in human-computer interaction and usability research: Ready to deliver the promises," in *The mind's eye*. Elsevier, 2003, pp. 573–605.

[129] G. Buscher, A. Dengel, R. Biedert, and L. V. Elst, "Attentive documents: Eye tracking as implicit feedback for information retrieval and beyond," *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 1, no. 2, p. 9, 2012.

[130] F. Roesner, T. Kohno, and D. Molnar, "Security and privacy for augmented reality systems," *Communications of the ACM*, vol. 57, no. 4, pp. 88–96, 2014.

[131] I. Arapakis, M. Lalmas, and G. Valkanas, "Understanding within-content engagement through pattern analysis of mouse gestures," in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. ACM, 2014, pp. 1439–1448.

[132] M. Singh, A. Nandi, and H. Jagadish, "Skimmer: rapid scrolling of relational query results," in *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*. ACM, 2012, pp. 181–192.

[133] P. E. Gill, W. Murray, and M. A. Saunders, "Snopt: An sqp algorithm for large-scale constrained optimization," *SIAM review*, vol. 47, no. 1, pp. 99–131, 2005.

[134] T. Siddiqui, A. Kim, J. Lee, K. Karahalios, and A. Parameswaran, "Effortless data exploration with zenvisage: an expressive and interactive visual analytics system," *Proceedings of the VLDB Endowment*, vol. 10, no. 4, pp. 457–468, 2016.

[135] M. Khan, L. Xu, A. Nandi, and J. M. Hellerstein, "Data tweening: incremental visualization of data transforms," *Proceedings of the VLDB Endowment*, vol. 10, no. 6, pp. 661–672, 2017.

[136] S. Amer-Yahia, B. Omidvar-Tehrani, J. Comba, V. Moreira, and F. Colque Zegarra, "Exploration of user groups in vexus," *arXiv preprint arXiv:2098926*, 2017.

[137] Z. Pang, S. Wu, G. Chen, K. Chen, and L. Shou, "Flashview: an interactive visual explorer for raw data," *Proceedings of the VLDB Endowment*, vol. 10, no. 12, pp. 1869–1872, 2017.

[138] E. Hoque and G. Carenini, "Convis: A visual text analytic system for exploring blog conversations," in *Computer Graphics Forum*, vol. 33, no. 3. Wiley Online Library, 2014, pp. 221–230.

[139] A. Sarvghad, M. Tory, and N. Mahyar, "Visualizing dimension coverage to support exploratory analysis," *IEEE transactions on visualization and computer graphics*, vol. 23, no. 1, pp. 21–30, 2017.

[140] W. Willett, J. Heer, and M. Agrawala, "Scented widgets: Improving navigation cues with embedded visualizations," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1129–1136, 2007.

[141] C. Collins, F. B. Viegas, and M. Wattenberg, "Parallel tag clouds to explore and analyze faceted text corpora," in *Visual Analytics Science and Technology, 2009. VAST 2009. IEEE Symposium on*. IEEE, 2009, pp. 91–98.

[142] K. Sharma, "Data visualization and visual analytics are not the same - know the difference," http://www.ascentt.com, 2017.

[143] B. Shneiderman, "The eyes have it: A task by data type taxonomy for information visualizations," in *Visual Languages, 1996. Proceedings., IEEE Symposium on*. IEEE, 1996, pp. 336–343.

[144] B. Saket, H. Kim, E. T. Brown, and A. Endert, "Visualization by demonstration: an interaction paradigm for visual data exploration," *IEEE transactions on visualization and computer graphics*, vol. 23, no. 1, pp. 331–340, 2017.

[145] J. Mackinlay, P. Hanrahan, and C. Stolte, "Show me: Automatic presentation for visual analysis," *IEEE transactions on visualization and computer graphics*, vol. 13, no. 6, 2007.

[146] J. Mackinlay, "Automating the design of graphical presentations of relational information," *Acm Transactions On Graphics (Tog)*, vol. 5, no. 2, pp. 110–141, 1986.

[147] M. J. Halvey and M. T. Keane, "An assessment of tag presentation techniques," in *Proceedings of the 16th international conference on World Wide Web*. ACM, 2007, pp. 1313–1314.

[148] C. Stolte, "Visual interfaces to data," in *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*. ACM, 2010, pp. 1067–1068.

[149] F. Dabek and J. J. Caban, "A grammar-based approach for modeling user interactions and generating suggestions during the data exploration process," *IEEE transactions on visualization and computer graphics*, vol. 23, no. 1, pp. 41–50, 2017.

[150] L. Lins, J. T. Klosowski, and C. Scheidegger, "Nanocubes for real-time exploration of spatiotemporal datasets," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, 2013.

[151] C. A. Pahins, S. A. Stephens, C. Scheidegger, and J. L. Comba, "Hashedcubes: Simple, low memory, real-time visual exploration of big data," *IEEE transactions on visualization and computer graphics*, vol. 23, no. 1, pp. 671–680, 2017.

[152] S. Kandel, R. Parikh, A. Paepcke, J. M. Hellerstein, and J. Heer, "Profiler: Integrated statistical analysis and visualization for data quality assessment," in *Proceedings of the International Working Conference on Advanced Visual Interfaces*. ACM, 2012.

[153] Z. Liu, B. Jiang, and J. Heer, "immens: Real-time visual querying of big data," in *Computer Graphics Forum*, vol. 32. Wiley Online Library, 2013, pp. 421–430.

[154] N. Bikakis, J. Liagouris, M. Krommyda, G. Papastefanatos, and T. Sellis, "graphvizdb: A scalable platform for interactive large graph visualization," in *Data Engineering (ICDE), 2016 IEEE 32nd International Conference on*. IEEE, 2016, pp. 1342–1345.

[155] M. De Choudhury, Y.-R. Lin, H. Sundaram, K. S. Candan, L. Xie, A. Kelliher *et al.*, "How does the data sampling strategy impact the discovery of information diffusion in social media?" *ICWSM*, vol. 10, pp. 34–41, 2010.

[156] S. P. Callahan, J. Freire, E. Santos, C. E. Scheidegger, C. T. Silva, and H. T. Vo, "Vistrails: visualization meets data management," in *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*. ACM, 2006, pp. 745–747.

[157] K. Huang, S. S. Bhowmick, S. Zhou, and B. Choi, "Picasso: exploratory search of connected subgraph substructures in graph databases," *Proceedings of the VLDB Endowment*, vol. 10, 2017.

[158] Z. Zhao, L. De Stefani, E. Zgraggen, C. Binnig, E. Upfal, and T. Kraska, "Controlling false discoveries during interactive data exploration," in *Proceedings of the 2017 ACM International Conference on Management of Data*. ACM, 2017, pp. 527–540.

[159] X. Yan and J. Han, "Closegraph: mining closed frequent graph patterns," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2003.

[160] C. Jin, S. S. Bhowmick, B. Choi, and S. Zhou, "Prague: towards blending practical visual subgraph query formulation and query processing," in *Data Engineering (ICDE), 2012 IEEE 28th International Conference on*. IEEE, 2012, pp. 222–233.

[161] Y. Fang, R. Cheng, S. Luo, and J. Hu, "Effective community search for large attributed graphs," *Proceedings of the VLDB Endowment*, vol. 9, no. 12, pp. 1233–1244, 2016.

[162] H. Ehsan, M. A. Sharaf, and P. K. Chrysanthis, "Muve: Efficient multi-objective view recommendation for visual data exploration," in *Data Engineering (ICDE), 2016 IEEE 32nd International Conference on*. IEEE, 2016, pp. 731–742.

[163] M. Vartak, S. Rahman, S. Madden, A. Parameswaran, and N. Polyzotis, "S ee db: efficient data-driven visualization recommendations to support visual analytics," *Proceedings of the VLDB Endowment*, vol. 8, no. 13, pp. 2182–2193, 2015.

[164] E. Zgraggen, R. Zeleznik, and S. M. Drucker, "Panoramicdata: Data analysis through pen & touch," *IEEE transactions on visualization and computer graphics*, vol. 20, no. 12, pp. 2112–2121, 2014.

[165] Y. Park, M. Cafarella, and B. Mozafari, "Visualization-aware sampling for very large databases," in *ICDE*. IEEE, 2016.

[166] U. Jugel, Z. Jerzak, G. Hackenbroich, and V. Markl, "Faster visual analytics through pixel-perfect aggregation," *Proceedings of the VLDB Endowment*, vol. 7, no. 13, pp. 1705–1708, 2014.

[167] E. Wu, L. Battle, and S. R. Madden, "The case for data visualization management systems: vision paper," *Proceedings of the VLDB Endowment*, vol. 7, no. 10, pp. 903–906, 2014.

[168] L. Jiang, P. Rahman, and A. Nandi, "Evaluating interactive data systems: Workloads, metrics and guidelines," *SIGMOD*, 2018.

[169] N. Kerracher, J. Kennedy, and K. Chalmers, "Tasks for temporal graph visualisation," *arXiv preprint arXiv:1402.2867*, 2014.

[170] J.-w. Ahn, C. Plaisant, and B. Shneiderman, "A task taxonomy for network evolution analysis," *IEEE transactions on visualization and computer graphics*, vol. 20, no. 3, pp. 365–376, 2014.

[171] D. E. Polkinghorne, "Language and meaning: Data collection in qualitative research." *Journal of counseling psychology*, 2005.

[172] T. Munzner, "A nested model for visualization design and validation," *IEEE transactions on visualization and computer graphics*, vol. 15, no. 6, pp. 921–928, 2009.

[173] K. A. Cook and J. J. Thomas, "Illuminating the path: The research and development agenda for visual analytics," *IEEE Computer Society, Los Alamitos, CA, United States (US).*, 2005.

[174] Z. Liu, Y. Wang, M. Dontcheva, M. Hoffman, S. Walker, and A. Wilson, "Patterns and sequences: Interactive exploration of clickstreams to understand common visitor paths," *IEEE transactions on visualization and computer graphics*, vol. 23, 2017.

[175] W. Mason and S. Suri, "Conducting behavioral research on amazons mechanical turk," *Behavior research methods*, vol. 44, 2012.

[176] A. I. Chittilappilly, L. Chen, and S. Amer-Yahia, "A survey of general-purpose crowdsourcing techniques," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, 2016.

[177] A. Kittur, E. H. Chi, and B. Suh, "Crowdsourcing user studies with mechanical turk," in *SIGCHI*. ACM, 2008, pp. 453–456.

[178] T. Blascheck, M. John, K. Kurzhals, S. Koch, and T. Ertl, "Va 2: a visual analytics approach for evaluating visual analytics applications," *IEEE transactions on visualization and computer graphics*, vol. 22, no. 1, pp. 61–70, 2016.

[179] "Oracle tpc benchmark," http://www.tpc.org, 2017.

[180] O. Erling, A. Averbuch, J. Larriba-Pey, H. Chafi, A. Gubichev, A. Prat, M.-D. Pham, and P. Boncz, "The ldbc social network benchmark: Interactive workload," in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. ACM, 2015, pp. 619–630.

[181] T. Milo and A. Somech, "Next-step suggestions for modern interactive data analysis platforms," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2018, pp. 576–585.

[182] P. Eichmann, C. Binnig, T. Kraska, and E. Zgraggen, "Idebench: A benchmark for interactive data exploration," *CoRR*, vol. abs/1804.02593, 2018.

[183] P. Eichmann, E. Zgraggen, Z. Zhao, C. Binnig, and T. Kraska, "Towards a benchmark for interactive data exploration." *IEEE Data Eng. Bull.*, vol. 39, no. 4, pp. 50–61, 2016.

[184] L. Battle, M. Angelini, C. Binnig, T. Catarci, P. Eichmann, J. Fekete, G. Santucci, M. Sedlmair, and W. Willett, "Evaluating visual data analysis systems: A discussion report," in *HILDA@SIGMOD*, C. Binnig, J. Freire, and E. Wu, Eds. ACM, 2018, pp. 4:1–4:6.



Behrooz Omidvar-Tehrani Behrooz Omidvar-Tehrani is a postdoctoral researcher in the University of Grenoble Alpes, France. Previously, he was a postdoctoral researcher at the Ohio State University, USA. His research is in the area of data management, focusing on interactive analysis of user data. Behrooz received his PhD in Computer Science from University of Grenoble Alpes in 2015. He has published in several international conferences and journals including VLDB, CIKM, KAIS, VLDB, ICDE, and EDBT.



Sihem Amer-Yahia Sihem Amer-Yahia is a CNRS Research Director at the University of Grenoble Alpes. Her interests are at the intersection of large-scale data management and user data exploration. Before joining CNRS, she was Principal Scientist at QCRI, Senior Scientist at Yahoo! Research and Member of Technical Staff at at&t Labs. Sihem is Editor-in-Chief of the VLDB Journal and has been on the editorial boards of TODS and the Information Systems Journal. She chaired VLDB 2018. Sihem received her Ph.D. in CS from Paris-Orsay and INRIA in 1999, and her Diplme dIngnieur from INI, Algeria.